****
*
 The FastDPeak program was compiled under Windows using c++ with CodeBlocks 10.05.
****
*

==========

= Files

==========

Unzip the "FastDPeak.zip" file, which will create folder mainly containing:
- a project file named "FastDPeak.cbp".
- a dataset folder named "data".
- a c++ file named "main.cpp" is the main function file.
- other c++ files

==========

= Environment configuration

==========

Step1:
- Download CodeBlocks in http://www.codeblocks.org/
- Download TDM-GCC-32 in http://tdm-gcc.tdragon.net/download/

Step2:
- Open CodeBlock: choose "setting" ->"compiler and debugger"->"ToolChain executables",and set the parameters like the "Figure 1" shows:

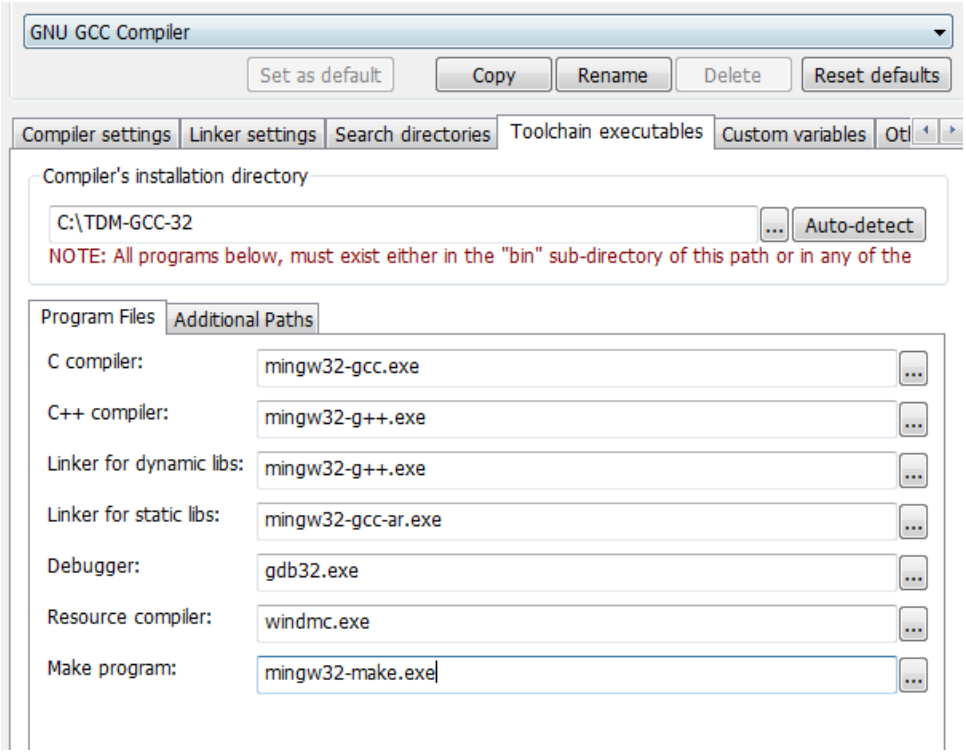

Figure 1: Configuring the "Toolchain executable" options

Step3:

Dataset Formation

The dataset should be given in a text file with the following formation:

- each line represents a point with *d* numbers, where *d* is dimension:

For instance, the first 20 lines of the sample dataset "agg.txt" are shown as below:

```
1.2     1.6
3       5
2       4.6
10      2
2.1     4.1
3.5     5.1
6.6     1
3.6     4.2
3       10.3
6.8     7.2
1.1     1.5
3.3     2.5
8.6     9.2
1.8     2.2
11.2    8.4
7.9     8.8
9.2     9.8
9.4     9
10.2    2
2.4     3.6
```

In this example, there are 2 numbers in each line: the first line represents the first point with the coordinates (1.2 1.6), and its id is 1. Similarly, the rest nine lines specify the coordinates of the point with id = 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,17,18,19,20 respectively.

Data Download

The data used in the paper can be downloaded from https://pan.baidu.com/s/1zqTmI7PPNeXfpcV_Xb9tuQ Password: bqxc, as Figure 2 shows.

Figure 2: Dataset downloading interface

-"new_KDD_data.txt" is "KDD99".
-"new_bio_train.txt" is "KDD04".
-"new_BigCross500K.txt" is "BigCross".
-"synthesis_1.txt" is "SYN1".
-"synthesis_2.txt" is "SYN2".
-"synthesis_3.txt" is "SYN3".

After downloading, the datasets should be put under the directory of "\FastDPeak\data", as the figure 3 shows.

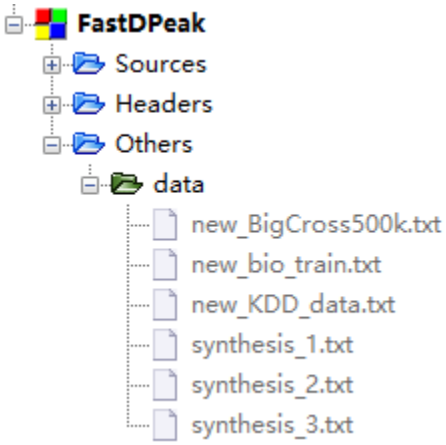

Figure 3: The directory of dataset

==========

An example of quick start

==========

Step1:

Open project "FastDPeak.cbp" in Codeblocks.

Step2:

Open "main.cpp",as figure 4 shows.

```
707  int main(int argc, char *const argv[]){
708
709      //Input data file
710      char *data_file_name = "data/agg.txt";
711      //Output log_files and result files
712      FILE* log_file = fopen("exp_log.txt","a+");
713      FILE* results= fopen("result.txt","a+");
714      FILE* cl_results = fopen("cl_results.txt","a+");
715
716      std::cout << "reading data...\n";
717      //data_size is the size of raw_data file,new_size is the size of new_data
718      int dim, data_size, new_size = 20;
719      //read data from data_file
720      float *raw_data = read_data(data_file_name," ",&dim, &data_size);
721      //generate new_data with index by new_size
722      float *data_with_index = Generate_data_with_index(raw_data,data_size,dim,new_size);
723      free(raw_data);
724      std::cout << "data read";
725
726      int K = 5,batch_num = 10,cl = 2,local_peak_threshold = 4;
727      float *dis_matrix = (float*)malloc(3*data_size*K*sizeof(float));
728      node_p *node_p_ptr = new node_p [data_size];
729      //perform Fast Density Peak Clustering
730      Fast_Density_Peak(K,data_with_index,new_size,batch_num,dim,local_peak_threshold
731                        ,log_file,results,cl,cl_results,dis_matrix,node_p_ptr);
732
733      free(data_with_index);
734      free(dis_matrix);
735
736      return 0;
737  }
```

Figure 4: Code screenshots of "main.app"

In line 720 : float *raw_data = read_data(data_file_name," ",&dim, &data_size);
-The first parameter data_file_name represents the origin dataset.
-The second parameter dim represents dim of the origin dataset.
-The third parameter data_size represents size of the origin dataset.

In line 722:
float *data_with_index = Generate_data_with_index(raw_data,data_size,dim,new_size);
-The first parameter raw_data represents the origin dataset.
-The second parameter data_size represents size of the origin dataset.
-The third parameter dim represents dim of the origin dataset.
-The fourth parameter new_size represents size of new dataset which we would like to classify.

In line 730-731:
Fast_Density_Peak(K,data_with_index,new_size,batch_num,dim,local_peak_threshold
,log_file,results,cl,cl_results,dis_matrix,node_p_ptr);
-The first parameter K represents the K of KNN.
-The second parameter data_with_index represents new dataset which we would like to classify.data_size represents size of the origin dataset.
-The third parameter new_size represents size of new dataset which we would like to

classify.
-The fourth parameter batch_num represents the number of batch in creating Covertree.
-The fifth parameter dim represents dim of the origin dataset.
-The sixth parameter local_peak_threshold represents threshold of local peak points.
-The sixth parameter log_file represents log file which record records of experiment.
-The seventh parameter results represents the tree of new dataset by FastDPeak.
-The eighth parameter cl represents the number of clusters.
-The ninth parameter cl_results represents the designate file in which FastDPeak will write the clustering result.
-The tenth parameter dis_matrix represents the matrix of distance.
-The eleventh parameter node_p_ptr represents the density and index of every point.

Step3:
-Press the "Build and run" button in CodeBlocks under release mode.
The result will perform like Figure 5:

```
F:\Code\FastDPeak\FastDPeak\bin\Release\FastDPeak.exe
reading data...
dim:2 n_size:20
data readbuilding tree for source data....
Runtime of building tree for source data is 0.000070 s
batch:2
generate queries...
Generate query data:0--10
building tree for query data....
Runtime of building tree for query data is 0.000026 s
K = 5, runtime of kNN is 0.000156 s
generate queries...
Generate query data:10--20
building tree for query data....
Runtime of building tree for query data is 0.000055 s
K = 5, runtime of kNN is 0.000205 s
K = 5,runtime of building covertree and KNN is 0.002293 s
Runtime of computing distance for query data: 0.000004 s
Runtime of finding initiative LDP for data 0.000001 s
Runtime of sorting density for source data is 0.000012 s
Runtime of PreProcessing local density peak by local D_peak threshold is 0.000000 s
Runtime of finding delta for root is 0.000001 s
Runtime of finding order end,runtime is 0.000000 s
Runtime of finding parent for local density peak is 0.000183 s
Runtime of determining final clusters 0.000001 s
Runtime of labeling cluster for data is 0.000001 s
Runtime of clustering for data is 0.003217 s

Process returned 0 (0x0)   execution time : 0.474 s
Press any key to continue.
```

Figure 5: The result of FastDPeak with the data "agg.txt"

==========

Output Format

==========

The clustering result is saved in "data\cl_results.txt", and the output formation is:

1.2 1.6 2

3	5	1
2	4.6	1
10	2	1
2.1	4.1	1
3.5	5.1	1
6.6	1	1
3.6	4.2	1
3	10.3	1
6.8	7.2	2
1.1	1.5	1
3.3	2.5	2
8.6	9.2	1
1.8	2.2	1
11.2	8.4	2
7.9	8.8	2
9.2	9.8	1
9.4	9	1
10.2	2	2
2.4	3.6	2

The first two columns store coordinates of all point, and the last column represents the cluster ID.

For example, the first line (1.2 1.6 2) means that the first point is classified into cluster 2. If cluster ID is -1, it means this point is a noise.

==========

= Experiments

==========

Some experiments we did in paper are listed in experiment_records.doc