\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The KNN\_BLOCK\_DBSCAN program was compiled under Windows using c++ with CodeBlocks 10.05.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

===================================================================================

**Files**

===================================================================================

These program mainly containing:

- a project file named "KNN\_BLOCK\_DBSCAN.cbp".

- a dataset folder named "data".

- a folder name “flann”, which saves the flann’s head file.

- a c++ file named “KNN\_BLOCK\_DBSCAN.cpp” is the main function file.

- other c++ files

===================================================================================

**Environment configuration**

===================================================================================

**Step1:**

- Download and compile FLANN.

FLANN can be downloaded from the following address:

[http://www.cs.ubc.ca/~mariusm/flann](http://www.cs.ubc.ca/_x0018_mariusm/flann)

After downloading and unpacking, the following files and directories should

be present:

- **bin**: directory various for scripts and binary files

- **doc**: directory containing this documentation

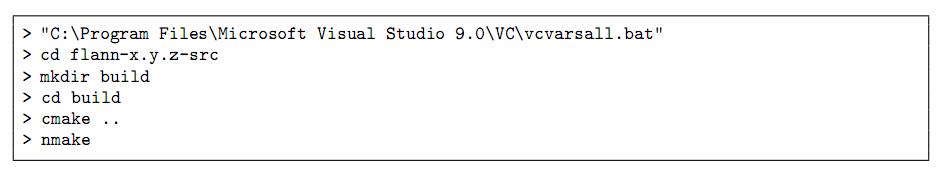
- **examples**: directory containing examples of using FLANN

- **src**: directory containing the source files

- **test**: directory containing unit tests for FLANN

To compile the FLANN library the “CMake” build system is required. Below

is an example of how FLANN can be compiled on windows.

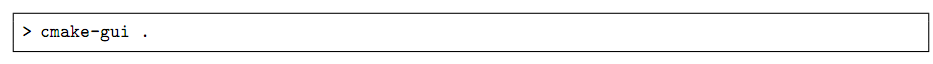


There are several compile options that can be configured before FLANN

is compiled, for example the build type (Release, RelWithDebInfo, Debug) or

whether to compile the C, Python or the MATLAB bindings. To change any of

this options use the cmake-gui application after cmake has finished (see Figure1).



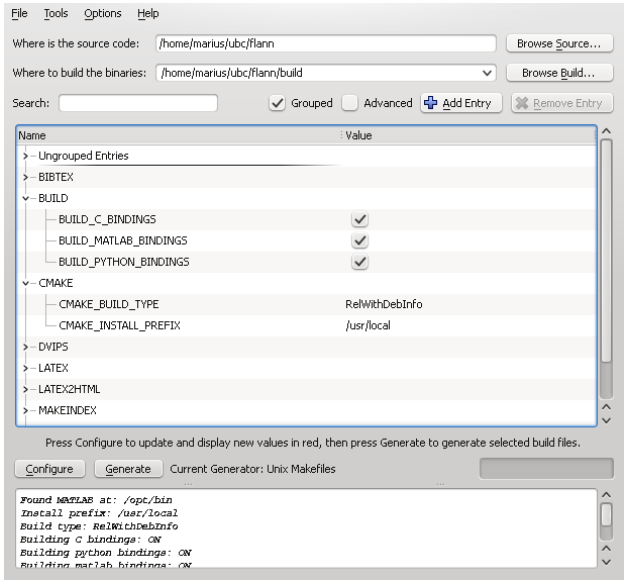


Figure 1: Configuring the FLANN compiler options

**Attention:If you want more information about compilering FLANN, you can visit** <https://github.com/mariusmuja/flann>

**Step2:**

- Download CodeBlocks in <http://www.codeblocks.org/>

- Download TDM-GCC-32 in <http://tdm-gcc.tdragon.net/download>/

**Step3:**

- Open CodeBlock: choose “setting” ->”compiler and debugger”->”ToolChain executables”,and set the parameters like the “Figure 2” shows:

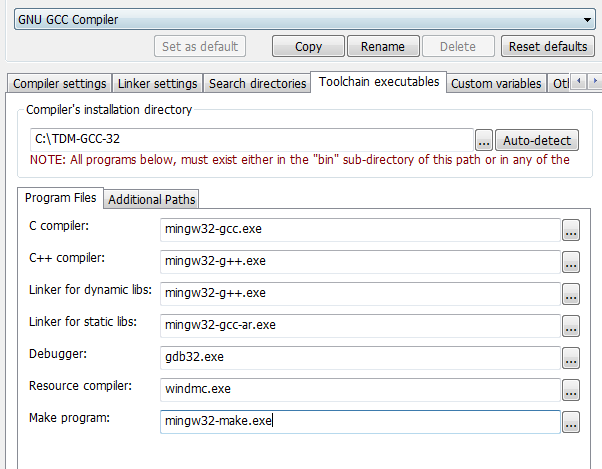


Figure 2: Configuring the “Toolchain executable” options

**Step4:**

- Open CodeBlocks : choose “setting” ->”compiler and debugger”->”Linker settings”, and add these FLANN lib files to “Link libraries” ,as “Figure 3” shows:

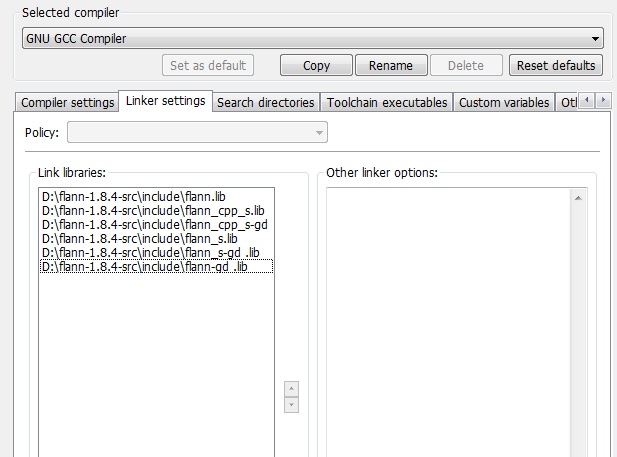


Figure 3: Configuring the “Linker settings” options

===================================================================================

**Dataset Format**

===================================================================================

The dataset should be given in a text file of the following format:

- In every line, there are $d+1$ numbers: the id of the point and the $d$ coordinate values, where the id is an integer from 0 to $n-1$ and each coordinate value is an integer.

For instance, the first 10 lines of the sample dataset "agg.txt"(whose data number is 789 and dimension is 2) are shown as below:

15.55 28.65

14.9 27.55

14.45 28.35

14.15 28.8

13.75 28.05

13.35 28.45

13 29.15

13.45 27.5

13.6 26.5

12.8 27.35

There are 2 numbers in each line: the first line represents that the first point whose Id is 1 with the coordinates (15.55 28.65). Analogously, the rest nine lines above specify the coordinates of the point with id = 2, 3, 4, 5, 6, 7, 8, 9,10 respectively.

===================================================================================

**An example of quick start**

===================================================================================

**Step1:**

Open project “KNN\_BLOCK\_DBSCAN.cbp” in Codeblocks.

**Step2:**

Open “KNN\_BLOCK\_DBSCAN.cpp”.

In line 22 : myClusterAnalysis.Init("data\\agg.txt", 1.35,7);

-The first parameter "data\\agg.txt" represents the origin dataset that you want to do clustering.

-The second parameter “1.35” represents Epsilon(eps).

-The third parameter “7” represents MinPts.

In line 29:myClusterAnalysis.WriteToFile("data\\result.txt");

-The parameter "data\\result.txt" represents the results file that you want to write in.

**Step3:**

Open “DataPoint.h”.

In line 4 :const int DIME\_NUM = 2;

-The value of the DIME\_NUM should be set as $d$(here the value is 2 because the dimension of “agg.txt” is 2).

**Step4:**

-Press the “Build and run” button in CodeBlocks and the code will be run.

The result will perform like Figure 4:

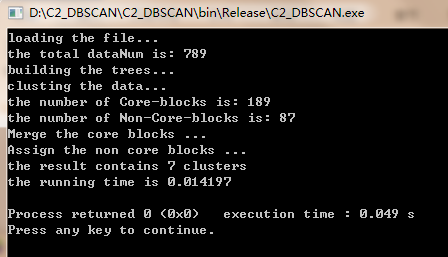


Figure 4: The result of KNN\_BLOCK\_DBSCAN with the data “agg.txt”

===================================================================================

**Output Format**

===================================================================================

The output of our program consists of one file(“data\\result”):

The first 10 lines in the file are as shown below:

15.55 28.65 1

14.9 27.55 1

14.45 28.35 1

14.15 28.8 1

13.75 28.05 1

13.35 28.45 1

13 29.15 1

13.45 27.5 1

13.6 26.5 1

12.8 27.35 1

The first line means that the point whose coordinate is (15.55 28.65) belongs to “cluster 1”. Analogously, the rest lines specify which cluster the point should belong to. And in particular, if cluster ID is -1, that represents this point is a noise point.