**Tutorial for calculating Boltzmann entropy of numerical raster data**

**Author**: Xinghua Cheng (mail: cxh9791156936@gmail.com)

The guide is designed to help users in the Ubuntu 18.04 system use the C++-based package that quickly calculates Boltzmann entropy of numerical raster data.

----------------------------------------------------------------------------------------------------------

Before installing GDAL for Linux, please open the terminal and then run the following commands one by one.

|  |
| --- |
| sudo add-apt-repository -y ppa:ubuntugis/ppa  sudo apt update  sudo apt upgrade |

After that, you should input the command as follows:

|  |
| --- |
| sudo apt install gdal-bin python-gdal python3-gdal libgdal-dev |

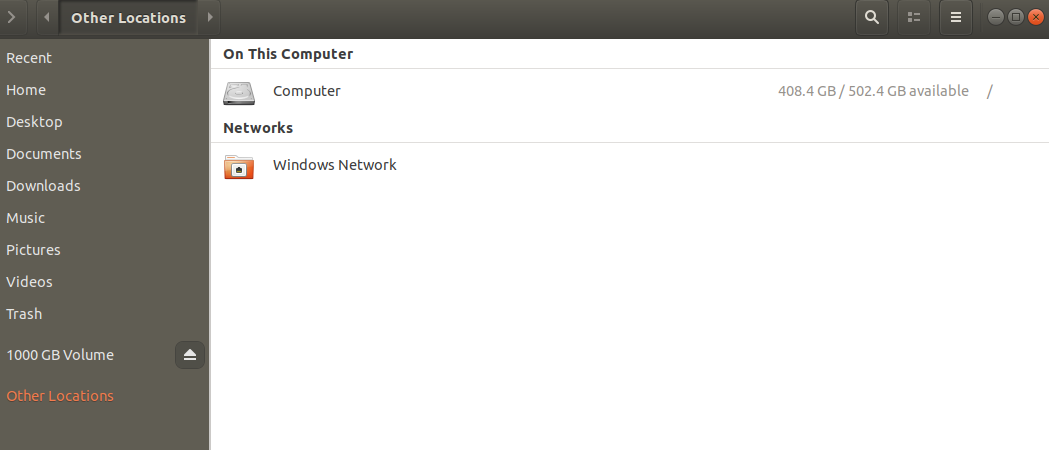
When the installation is completed, you can run “gdalinfo --version” to check out the version of gdal installed in your system (see the Fig 1).

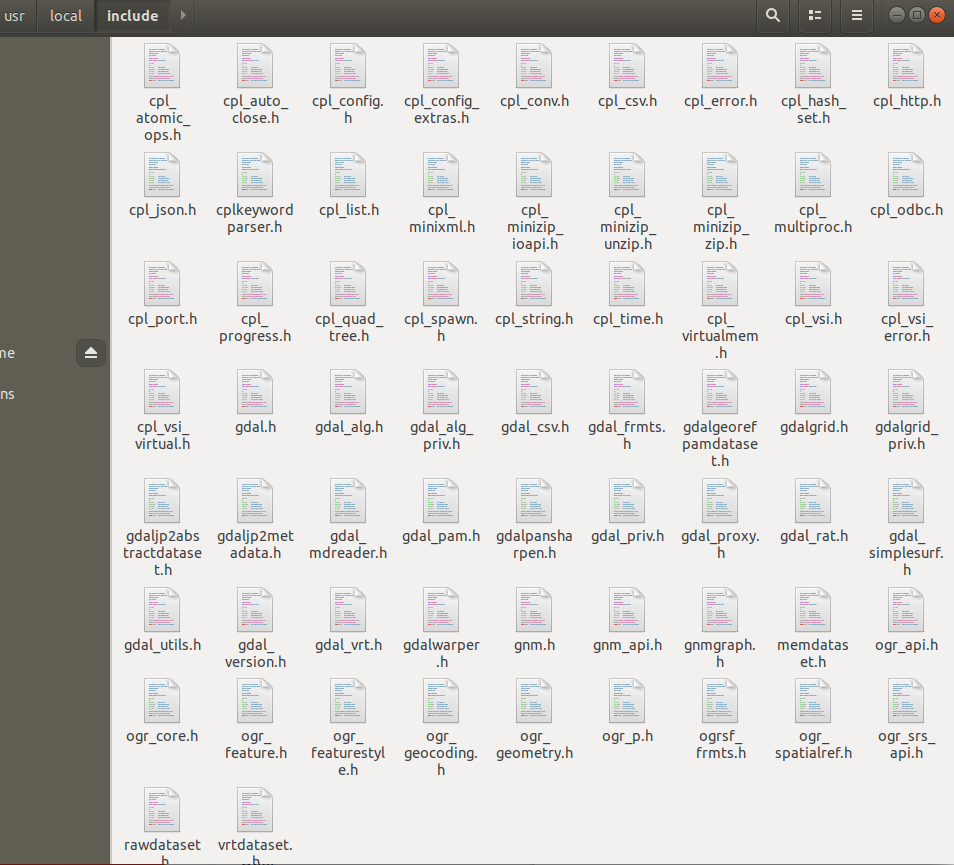
Text

Description automatically generated

**Fig 1**. The successful completion of GDAL installation.

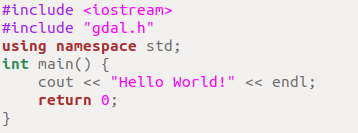
[Warming tips: you can run “sudo su” to change to the root. In this case, all GDAL libraries will be automatically installed in the directory of “usr/local” (see Fig 2), indicating that you do not need to set up the environmental variable or path.]



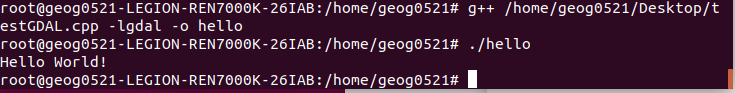


**Fig 2.** Click “other locations” and then click “Computer” on the right side to see the director of “usr/local”**.**

To test the GDAL, you should install G++ by the command of “sudo apt-get install g++”. Then, you can input the command of “g++ /home/geog0521/Desktop/tetstGDAL.app -lgdal -o hellofunction” to get the compiled function called “hello”. Afterwards, you can input “./hello” to get the output-”hello world!” (see Fig 4).



**Fig 3.** The codes in the “testGDAL.app”.



**Fig 4**. Results of running the compiled function of “hello”.

Next, you should compile the C++ file (i.e., cppBE\_0414\_test.cpp) of calculating Boltzmann entropy with the following command:

|  |
| --- |
| g++ /home/geog0521/Desktop/BECplusplus/cppBE\_0414\_test.cpp -lgdal -o /home/geog0521/Desktop/BECplusplus/BECplusplus\_0414\_test -lgomp |

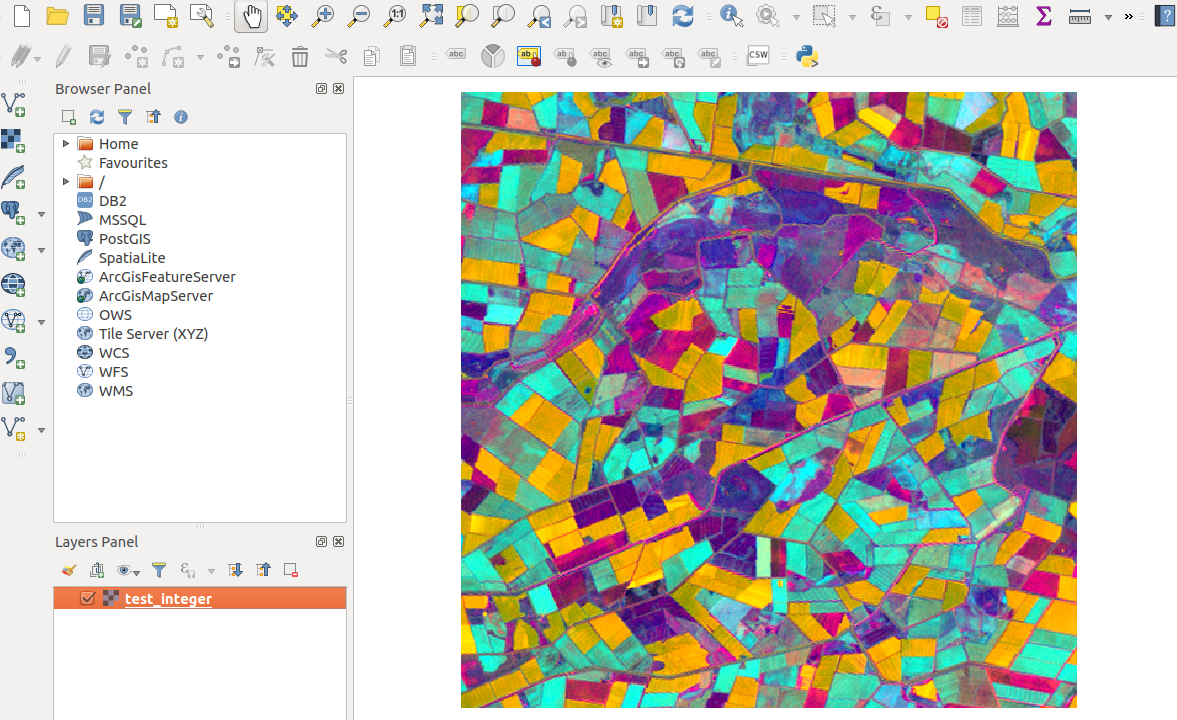
The compiled file is “BECplusplus\_0414\_test”.

Then, you can use “BECplusplus\_0414\_test” to calculate the Boltzmann entropy of an image (test\_integer.tif, see Fig 5) with the command as follows:

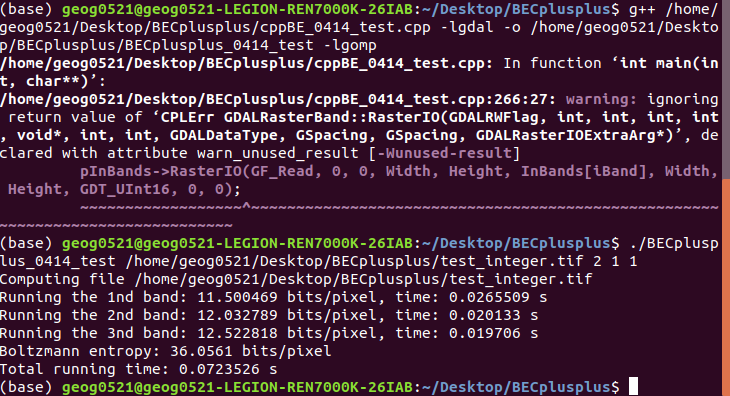
|  |
| --- |
| ./BECplusplus\_0414\_test /home/geog0521/Desktop/BECplusplus/test\_integer.tif 2 1 1 |

where parameter “2” represents the base for , the first “1” denotes normalization, the second “1” means that the output Boltzmann entropy value will be the sum of Boltzmann entropy of each band.

The computation results can be found in Fig 6.

****

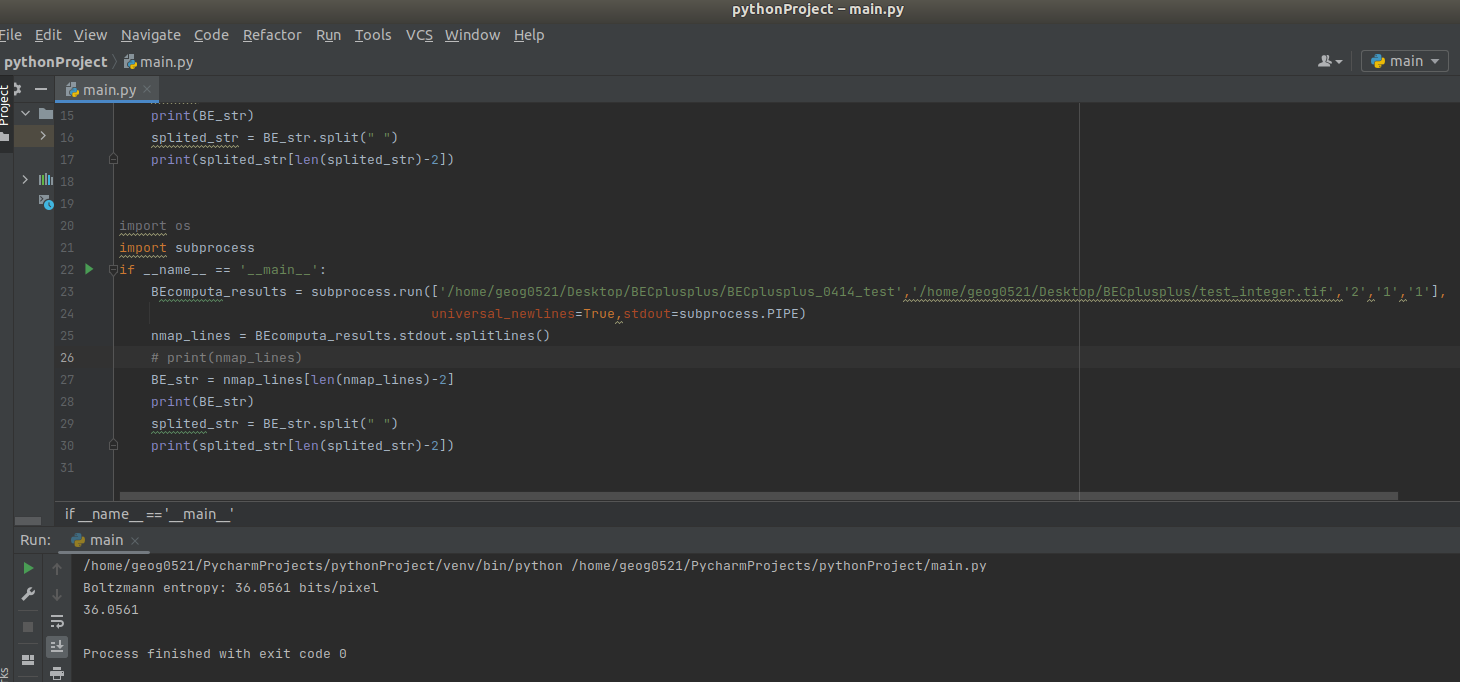
**Fig 5.** Test\_integer.tif with three bands.



**Fig 6.** The computation results of “Test\_integer.tif”.

In the python environment, you can use the following codes to calculate the relative Boltzmann entropy of “Test\_integer.tif” and get the returned Boltzmann entropy value.

|  |
| --- |
| import subprocess  if \_\_name\_\_ == '\_\_main\_\_':  BEcomputa\_results = subprocess.run(['/home/geog0521/Desktop/BECplusplus/BECplusplus\_0414\_test','/home/geog0521/Desktop/BECplusplus/test\_integer.tif','2','1','1'],universal\_newlines=True,stdout=subprocess.PIPE)  nmap\_lines = BEcomputa\_results.stdout.splitlines()  # print(nmap\_lines)  BE\_str = nmap\_lines[len(nmap\_lines)-2]  print(BE\_str)  splited\_str = BE\_str.split(" ")  print(splited\_str[len(splited\_str)-2]) |



**Fig 6.** The computation results of “Test\_integer.tif” by using Python.