Evaluating Virtual Screening Methods: Good and Bad Metrics for the "Early Recognition" Problem – Supporting Information

We provide the source code to calculate the Boltzmann-enhanced discrimination of receiver operating characteristic (BEDROC) metric for ranking results. We first describe the input format that the tools use and then give more detail on each tool. The Python (www.python.org) tool does not need compilation, is sufficiently fast and easy to modify. It is however mainly focused on BEDROC calculation. On the other hand, the C++ toolkit is more elaborate and can be used to do statistical analysis and many kinds of metric calculations. The provided files need to be renamed according to the instructions given below.

File renaming from ACS supporting information

ci600426e-File001.txt → Enrichment.cc
ci600426e-File002.txt → Metric.cc
ci600426e-File003.txt → Metric.h
ci600426e-File004.txt → bedroc.cc
ci600426e-File005.txt → Enrichment.h
ci600426e-File006.txt → bedroc.py
ci600426e-File007.txt → Enrichment.py
ci600426e-File008.txt → test1.txt
ci600426e-File009.txt → test2.txt
ci600426e-File010.txt → test3.txt
ci600426e-File011.txt → test4.txt

Ranking result input format

The input files test1.txt, test2.txt, test3.txt and test4.txt show the format read by the programs provided. An example:

177	1
777	2
1235	3
1487	4
1578	5
1632	6
1819	7
2341	8
2614	9
2946	10
5000	10

The first column indicates the rank position of each retrieved active and the second column is the cumulative count of actives. The last line is used only to identify the total number of compounds (N=5000) and the number of actives (n=10). If not all the actives are kept (sometimes they are rejected even before they are scored), it is possible that the cumulative count does not reach the total number of actives, but the last line is still used to specify N and n.

Python

The Python source code is focused on the calculation of the BEDROC metric. The file bedroc.py exemplifies the use of the Enrichment class. The usage is shown in the ***Test*** section.

C++

Two main classes are provided: Metric and Enrichment. The former is an abstract class from which metric function objects are derived. The latter is the ranking information holder and is made to manipulate the data. The program bedroc.cc exemplifies the use of the C++ toolkit in providing a program to calculate the BEDROC metric. The usage is shown in the ***Test*** section.

Here is how to compile bedroc with GNU g++ compiler:

```
g++ Enrichment.cc Metric.cc bedroc.cc –o bedroc
```

This source code has been successfully compiled on three different platforms including Windows (Microsoft Visual C++ 2005 and cygwin GNU g++ compilers), Suse Linux on IBM workstation with Opteron dual processors (pgCC and GNU g++ compilers) and AIX power5 and power4 processors (xlC compiler). We can also provide a Win32 executable upon request.

Test

Four test files are provided to verify the compiled C++ bedroc executable. Here are the expected BEDROC scores:

```
bedroc 20.0 test1.txt test2.txt test3.txt test4.txt

test1.txt 0.0738885
test2.txt 0.285047
test3.txt 1
test4.txt 0.796369

python bedroc.py 20.0 test1.txt test2.txt test3.txt test4.txt

test1.txt 0.074
test2.txt 0.285
test3.txt 1.000
test4.txt 0.796
```

In both cases, the command line first argument is the BEDROC exponential factor (positive). The recommended value is 20.0.