

PROMETHEUS

software guide

Introduction

The Prometheus application provides an overall PBT (persistence, bio-concentration, toxicity) assessment based on QSAR modelling, combined in a final score useful for ranking compounds.

The application is based on the conceptual scheme developed within the Prometheus project of the UBA (Umwelt Bundesamt, German Federal Environment Agency) and fully reported in the following publication:

F Pizzo, A Lombardo, A Manganaro, C Cappelli, M Petoumenou, F Albanese, A Roncaglioni, M Brandt, E Benfenati, “**Integrated in silico strategy for PBT assessment and prioritization under REACH**”, *Environmental Research* 151, 478-492

The application relies on a battery of QSAR models, combined in a workflow resulting in a final prediction for the P, B and T property of each compound, provided with a reliability value. The P, B and T prediction are then transformed into normalized scores, which are combined into a global PBT score; a PB score is also provided, where the T property is not taken into account (for vPvB assessment). The used models come from the VEGA platform (see www.vega-qsar.eu for further details) and are directly calculated inside Prometheus; some external models can be further added, providing their predictions inside the input file (see the “input” section for details). The application also provides a LogP prediction, which is not directly used for the final PBT/PB score. This logP prediction is a slightly different implementation of the workflow explained in the mentioned paper, and takes into account only the three models available in VEGA (Meylan, AlogP, MlogP).

The application is developed in Java and can be run on every platform supporting the Java Runtime Environment, version 7 (see www.java.com for further details).

How to use

Prometheus has both a graphical user interface (GUI) and a command line interface (CLI).

To start the GUI version, it is possible to double-click from on the *Prometheus.jar* file, or to run it from the command line / shell environment of your computer with the following command:

```
java -jar Prometheus.jar
```

The GUI application requires to set an input file and to choose a destination file where the results will be saved. Then, all calculations will be started by clicking the “Calculate” button. In the lower part of the form, a log panel will report the information (and possible warnings/errors) about the calculation.

To start the CLI version, from the command line / shell environment of your computer, run the application with the following command and with at least one parameter (note that if no parameters are provided the GUI will start):

```
java -jar Prometheus.jar [parameters]
```

The available parameters are the following:

-l or --log-stdout	Send log to STDOUT (if not provided, STDERR is used by default)
-h or --help	Show the help with the full list of parameters
-input <file>	Use the given file for input
-output <file>	Use the given file for output (if not provided, STDOUT is used by default)
-s or --silent	Silent mode (no process log is provided)

Input format

The Prometheus application software requires as input a plain text file containing the SMILES encoding of a molecule for each line. No header lines should be added.

Additional fields can be added on each line, separated by tab characters. These fields should be used when additional information about available experimental values and/or external QSAR models results should be considered for the PBT assessment of the given molecule. Each additional field should be given in the form:

<key>:<value>

where the key identify what kind of data is going to be added, and the value reports the numerical value of that data. Currently available keys are reported in the following table:

Key	Meaning
p_wat_exp	Persistence experimental data for water compartment
p_soil_exp	Persistence experimental data for soil compartment
p_sed_exp	Persistence experimental data for sediment compartment
bcf_exp	BCF experimental value
t_exp	Toxicity experimental value
t_test_exp	Toxicity experimental value found by EPA T.E.S.T. software
t_test_pred	Toxicity predicted value from EPA T.E.S.T. software
t_eco_pred	Toxicity predicted value from ECOSAR
logp_exp	LogP experimental value

For example, a valid input line would be:

```
C1=CC(C2C1C3(C(=C(C2(C3(C1)C1)C1)C1)C1)C1 t_test_exp:0.0661 t_test_pred:0.13
```

The above line means that for the molecule with the given SMILES, the software should consider in its assessment also a toxicity experimental value of 0.0661 found in T.E.S.T. and a toxicity prediction of 0.13 performed by T.E.S.T.

Note that it is acceptable to have keys without values, in this case the key is not considered while the input line is still valid. For instance the following line is a valid input line:

```
C1=CC(C2C1C3(C(=C(C2(C3(C1)C1)C1)C1)C1)C1 t_test_exp: t_test_pred:0.13
```

Note that experimental values for the three persistence compartment should be given using the following possible classes:

nP	Non persistent
nP/P	Non persistent / Persistent
P/vP	Persistent / Very Persistent
vP	Very Persistent

The meaning of these four classes is explained in the paper mentioned above and it is further detailed in the publication:

F Pizzo, A Lombardo, M Brandt, A Manganaro, E Benfenati, “**A new integrated in silico strategy for the assessment and prioritization of persistence of chemicals under REACH**”, *Environment international* 88, 250-260

Output format

The output (both on STDOUT and on a file) consists of a plain text, with tab-separated values. It reports, for each compound given as input, the following values:

no.	Progressive number of the compound
SMILES	SMILES of the compound (as processed by VEGA platform, it may be written in a different way from the original input compound)
LogP	LogP predicted value
LogP rel.	LogP prediction reliability (from 0 to 1, where 1 is the max reliability)
P	Persistence predicted value (as one of the classes: nP, nP/P, P/vP, vP)
P rel.	Persistence prediction reliability (from 0 to 1, where 1 is the max reliability)
B	BCF predicted value
B rel.	BCF prediction reliability (from 0 to 1, where 1 is the max reliability)
T	Toxicity predicted value (given as mg/l)
T rel.	Toxicity prediction reliability (from 0 to 1, where 1 is the max reliability)
Score P	Persistence normalized score
Score B	BCF normalized score
Score T	Toxicity normalized score
PBT	Overall PBT score
PB	Overall PB score

Disclaimer and contacts

The present software has been developed under a grant by Umwelt Bundesamt (UBA). UBA holds the intellectual property of the software and of the methodology hereby implemented. UBA releases the software under a GPL3 (GNU General Public License version 3) license.

The software has been developed by the following consortium:

Kode Chemoinformatics (Italy)

Istituto di Ricerche Farmacologiche Mario Negri (Italy)

KnowledgeMiner Software (Germany)

For questions and technical assistance please contact:

chm@kode-solutions.net