QSARSetBuilder

1 Introduction

The QSARSetBuilder enables you to clean ChEMBL data and rank order the assays within. This is done by treating each assay as a block of data, combining many sets of these blocks, and building and testing a low complexity model from each set. This produces a report containing how often each assay appears in a good set relative to a poor set.

This program was developed by Travis Hesketh during his undergraduate industrial placement. It is available under the GNU GPL v3 license (https://www.gnu.org/licenses/gpl-3.0.en.html).

The following sections describe the requirements necessary for setting up and using QSARSetBuilder.

2 Requirements

Python 3.6

RDKit

MolVS

scikit-learn

chemlistem (optional, required for substrate analysis)

tensorflow **1.3** (optional, required by chemlistem for substrate analysis)

h5py (optional, required for substrate analysis)

F5-6 Blenheim House, Cambridge Innovation Park
Denny End Road
Cambridge, CB25 9PB, UK

Tel: +44 1223 815900
Fax: +44 1223 815907

Email: info@optibrium.com
Website: www.optibrium.com

3 Installation

The easiest way to install the dependencies is to install the Anaconda 3 Python distribution and use conda to set up an environment with RDKit.

To install the other dependencies, use Python's pip package manager inside this environment.

On Linux and macOS:

```
pip install molvs scikit-learn matplotlib
```

Optionally:

```
pip install chemlistem tensorflow==1.3 h5py
```

On Windows:

```
python -mpip install molvs scikit-learn matplotlib
```

Optionally:

```
python -mpip install chemlistem tensorflow==1.3 h5py
```

To install QSARSetBuilder, download the zip file and extract it to:

On Linux and macOS:

```
~/QSARSetBuilder
```

On Windows:

```
C:\Users\Username\QSARSetBuilder
```

Make sure the .py files are in this folder and not in a subdirectory.

4 Usage

To use QSARSetBuilder, you need to run it from the command line using the Anaconda environment's version of Python (**not** the system executable). This can be done by activating the environment or using absolute paths. This section assumes that the default anaconda installation directory (~/anaconda3 or C:\Users\Username\Anaconda3 on windows) and RDKit environment name (my-rdkit-env) were used.

To activate the source on Linux or macOS (**on macOS, use pythonw instead of python**):

```
cd ~/anaconda3/bin

source activate my-rdkit-env

cd ~/QSARSetBuilder

python qsarsetbuilder.py /path/to/chembl/data.txt
```

On Windows:

```
activate my-rdkit-env

cd C:\Users\Username\QSARSetBuilder\qsarsetbuilder.py

python qsarsetbuilder.py C:\path\to\chembl\data.txt
```

To use absolute paths on Linux or macOS (**on macOS, use pythonw instead of python**)

```
~/anaconda3/envs/my-rdkit-env/bin/python ~/QSARSetBuilder/qsarsetbuilder.py /path/to/chembl/data.txt
```

On Windows:

```
C:\Users\Username\Anaconda3\envs\my-rdkit-env\python
    C:\Users\Username\QSARSetBuilder\qsarsetbuilder.py C:\path\to\chembl\data.txt
```

This will by default generate 1000 sets from the data and test them using regression models. See **4.3 Configuration** (or use the --help flag on the command line) for more options.

4.1 Expected Input

ChEMBL data for **one** target in a tab-delimited or comma-delimited format (e.g. all data for CHEMBL240, hERG). The following fields must be present in the file:

Required fieldnames are *CANONICAL_SMILES, STANDARD_TYPE, PCHEMBL_VALUE* and *ASSAY_CHEMBLID*, where:

- *CANONICAL_SMILES* is the SMILES string for a given molecule. These are automatically converted to a standardised format.
- *STANDARD_TYPE* is the type of measurement at the target (EC50, IC50, Ki, etc.). Only certain types are kept, these are given in the configuration options (see configuration).
- *PCHEMBL_VALUE* is the negative base 10 log of the activity value for the target in molar concentration (e.g. if the *STANDARD_TYPE* of the value is IC50, this will be the pIC50 value).
- *ASSAY_CHEMBLID* is the identifier for the assay the result was measured in. In ChEMBL, an 'assay' is a series of measurements from one paper using the same assay conditions.

Optional fieldnames are *RELATION, STANDARD_VALUE, STANDARD_UNITS* and *DESCRIPTION*, where:

- *RELATION* is an equality operator (<, <=, >, >=, ~ or =). If this is not '=', the value is discarded.
- *STANDARD_VALUE* is the non-logged measured value.
- *STANDARD_UNITS* are the molar units of the measured value. These are usually nM.
- *DESCRIPTION* is a string describing some important details of the assay conditions.

4.2 Output

The software outputs a directory (*modelname_analysis*) containing the following files (where 'modelname' is given by the name of the input file – for example, if the input file is *hERG.csv*, 'modelname' is *hERG*):

- A cleaned version of the input file after standardisation (.CSV)
- A cached binary version of the information contained in the clean file. (.CBSRT)

- A directory with a name given by the ISO 8601 time string (e.g. *20180309T105903Z*) containing run information:
    - The run log file (.LOG)
    - If run with chemlistem, this will conatain a subdirectory for each substrate/STANDARD_TYPE combination (e.g. *midazolam_IC50*). If not, using chemlistem, the directories will be called all_STANDARD_TYPE :
        - Report file (.TXT)
        - Assay similarity matrix (.CSV)
        - Set size distribution histogram (.PNG)

4.3 Configuration Options

4.3.1 Hardcoded configuration options

These configuration options can be changed by changing the qsarsetbuilder.py file:

Option Name	**Value**	**Description**
CACHE_DIR	~/.qsarsetbuilder (Linux/macOS) C:\Users\Username\ .qsarsetbuilder (Windows)	Path to the directory where the cached descriptors, fingerprints and non-canonical to canonical smiles transformations are kept. The model file for chemlistem is also stored in this directory. If this doesn't exist, it is created at runtime (and on Windows, is hidden using a Windows API call)
PROPERTIES_TO_CACHE	IC50, EC50, ED50, AC50, Ki, Kd	The STANDARD_TYPES to read in.
FALSE_MATCHES	amino, aminoacid, alkaloids, pharmacol, pubchem, concentration, compounds, fluorometer, phosphate, acetonitrile, nadph	Some false matches picked up by chemlistem. These are unlikely to be meaningful substrates.
SUBSTRATE_MISTAKES	bezyl: benzyl quiniline: quinoline	Misspellings of chemical names commonly encountered in ChEMBL.

4.3.2 Command line options

Flag	Option Type	Description and Default Value
--models, -m	c, r, cr, rc	Models to build and test. If c is present, classification models will be built. If r is present, regression models will be built. Default r
--split, -s	Decimal or integer	The value for the Active/not active cut off for classification models. Default value: 6 (corresponds to 1 µM activity)
--num-sets, -n	Integer	Number of sets to generate. If this is greater than the maximum possible number of combinatorial sets, all possible combinatorial sets will be built. Default 10000
--num-cores -c	Integer	Number of CPU cores to use for multiprocessing. Default all
--cont-after, -o	Integer	This flag is used for early termination if no good sets have been generated in this number of preceding sets. Default 1000
--max-length. -l	Integer	Maximum size of each assay set (in number of compounds). Default 2000
--properties, -p	list of strings	List of STANDARD TYPES (separated by spaces) to test. These must be present in PROPERTIES_TO_CACHE. Default PROPERTIES_TO_CACHE
--encoding	string	Python encoding to use for the CSV/TSV files. StarDrop uses windows-1252

		Default utf-8
--priority	idle, below_normal, normal, above_normal. high	Priority of processes spawned using multiprocessing. Values above_normal and high have no effect on Linux and macOS systems as root access is required to reduce the nice value of processes. Default below_normal
--excl-terms	list of strings	Terms to exclude. If these terms are present in the assay's description, the assay is not added. Default high-throughput, insect, baculovirus, oocytes
--excl-assays	list of strings	Assay IDs to exclude. These assays will not be added. Default none
--substrates	Flag - no options.	Use chemlistem to analyse substrates and perform separate runs for each substrate.
--no-oor	Flag - no options.	Don't use STANDARD_VALUE and STANDARD_UNITS to calculate a value if PCHEMBL_VALUE is missing.
--no-cache	Flag - no options.	Don't use the cache for this run.
--purge-cache	Flag - no options.	Clear the cache this run.
--test-set, -t	Path to test set file	See **4.4 External Test Set** for details.

4.4 External Test Set

By default, QSARSetBuilder evaluates models by first using a traditional training/validation split and then by using 5-fold cross validation on the training set. Models which have an R^2 or MCC score of >0.6 in both tests are said to be 'good'.

However, **if you already have** a manually curated test set (an *external* test set) for the target of interest, this can be used in place of the cross-validation test. This is done by specifying the file using the '--test-set' or '-t' option.

If this flag is specified, the compounds present in the external test set will be removed from the assay sets before the training/validation split evaluation. The training set will then be used to build a model with which to predict the values for the validation set and this external test set.

This external test set requires the *CANONICAL_SMILES* and *PCHEMBL_VALUE* fields only and should be a CSV/TSV file with the same encoding as the ChEMBL file.

5 Implementation Details

SMILES standardisation

- SMILES strings are read in using RDKit and those which fail to be read in are discarded
- Stereochemistry is removed in RDKit:
    - Chem.RemoveStereoChemistry(mol)
- Salts are removed in RDKit:
    - Chem.SaltRemover.SaltRemover().StripMol(mol, dontremoveeverything=True)
- MolVS is used to standardise the tautomer and group representations:
    - Standardizer().tautomer_parant(mol)
- The result is a canonical, non-isomeric SMILES string which has been normalised for tautomers and functional group representations
- **This resulting SMILES string is treated as a unique molecule**

Input filtering

- If *STANDARD_TYPE* is not one of the types listed for the --properties command line option or the default PROPERTIES_TO_CACHE, the assay is not included in the run ('not added').
- If PCHEMBL_VALUE is not present, it is calculated from *STANDARD_VALUE* and *STANDARD_UNITS* if no_oor is not specified in the command line options.
    - *PCHEMBL_VALUE* is often missing if its value would be < 4, as ChEMBL considers this 'out of range'.
    - If the no_oor command line option **is** specified, the individual result is discarded.
- If *ASSAY_ID* is listed for the --excl_assays command line option, the assay is not added.
- If *RELATION* is not '=', the individual result is discarded.
- *DESCRIPTION* based filtering:
    - The assay description often contains probe substrates and cell types and may include additional details such as measurement temperature or time but could include neither.
    - If these strings contain probe substrates, chemlistem (see the –substrates command-line option) can be used to pick out chemical names and perform separate runs for each substrate.
    - If any of the terms listed for the --excl_terms command line option are in the description, the assay is not added.
- If the assay has 2 compounds or fewer, or is a subset of another assay, it is not added.
    - 'Subsets' are assays which contain the same exact same SMILES: value pairs as another assay but are missing some additional measurements. This usually occurs where a newer paper which contains more measurements repeats older results.
    - In cases where assays are exact duplicates of others, only one assay is added.