

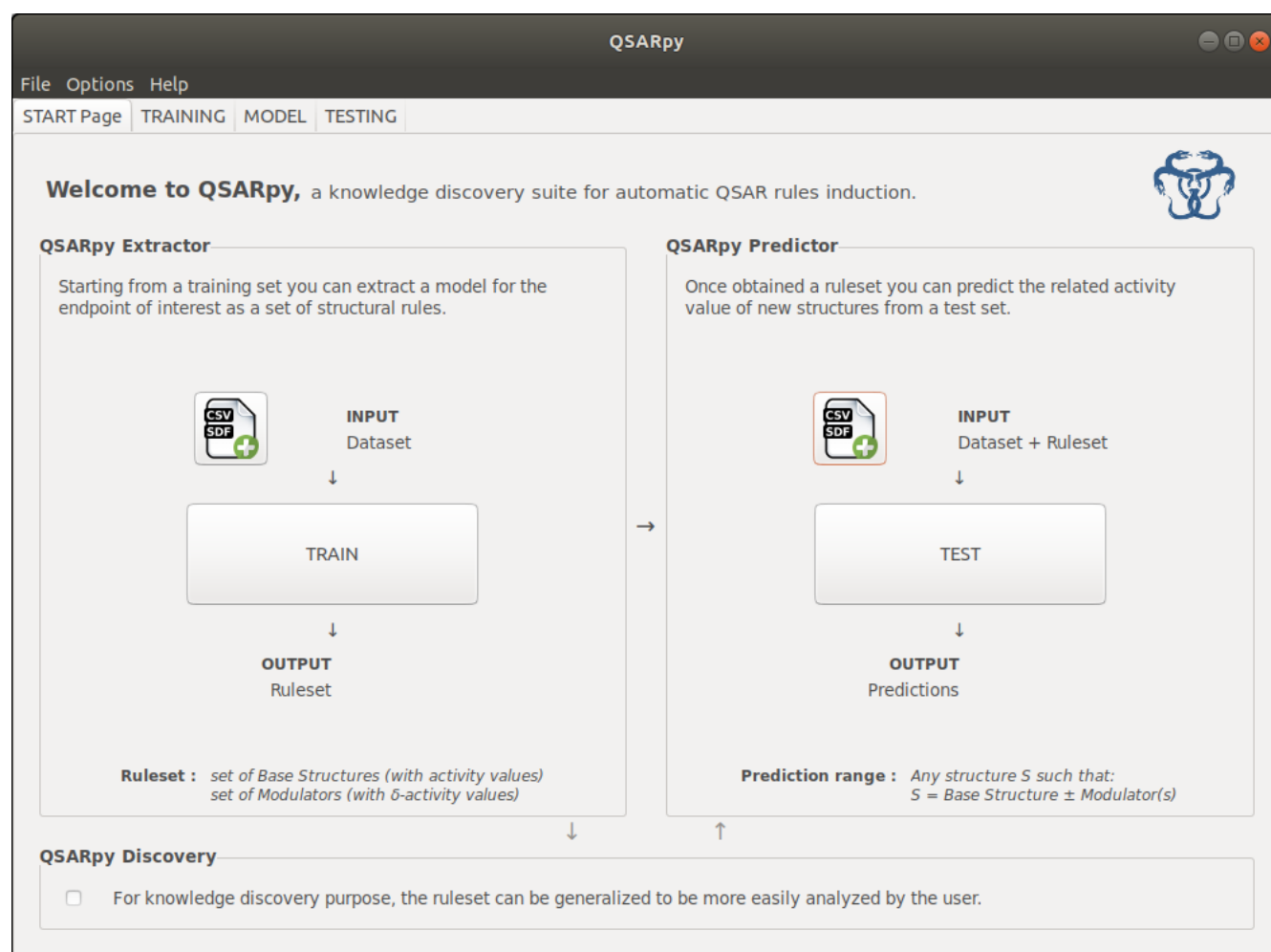
Welcome to QSARpy version 1.1 !

QSARpy is a knowledge discovery suite for automatic QSAR/QSPR rules induction.

By running **QSARpy Extractor** with a training set, you can extract a model for the endpoint of interest as a set of structural rules (the *ruleset*).

By running **QSARpy Predictor** with a test set, you can use the *ruleset* to predict the activity/property value of new structures.

Moreover, you have at disposal a set of tools to validate and analyze the model with absolute transparency and total human readability.



For simplicity, the rest of the manual is focused on the QSAR case, but the word 'activity' can always be replaced by 'property' (or any other numerical value that can be related to the molecular structure).

1. Start Page

This is the navigation page: from here, you can load a training set and start a new **QSARpy Extractor** session, or load a test set and start a new **QSARpy Predictor** session.

These two QSARpy tools can be run as stand-alone applications by means of the *Load/Save* functions in the **File** menu. Moreover, a default workflow is proposed :

QSARpy Extractor > QSARpy Predictor

and the output of the former is sent as input to the latter.



Load a dataset:

Choose a file containing the molecular structure notations.

Accepted file formats are:

- CSV (Comma Separated Values) containing the SMILES structural line notation,
- SDF (Structure Data File).

If the dataset is intended to be used as training set, each structural notation must be paired with the relative activity value.

CSV files can be exported by any spreadsheet editor like MS Excel or LibreOffice Calc.

Once a dataset has been loaded, hit **<TRAIN>** to start a **QSARpy Extractor** session or hit **<TEST>** to start a **QSARpy Predictor** session.

New v1.1: QSARpy Discovery

By checking this box, a new panel is inserted in the standard workflow of the program:

QSARpy Extractor > QSARpy Discovery > QSARpy Predictor

After a *ruleset* has been extracted, it can be generalized and made it simpler to be easily analyzed by the user. Then the generalized model is ready to be used for prediction.

2. QSARpy Extractor

To start a new session load a training set from the file menu: **File > Load Training set...**

See also “Load a dataset” for file format details (Chapter 1: START Page).

SET TRAINING SET HEADERS

Specify the required field headers in the training set file:

- **ID:** CAS number or any other ID suitable to identify the molecular structure
- **SMILES** (CSV file format only)
- **Prediction Target:** the activity/property numerical value to be predicted

The 'ID' field is optional. However, models extracted by QSARpy refer to the training structures: setting an appropriate structural identifier can facilitate reading the rationale behind predictions.

Hit **<Load>** to load the molecular structures of the dataset as training structures.

SET ERROR THRESHOLD:

Set the desired upper threshold for the prediction error of the single modulator.

All modulators showing a higher error during the training phase will be discarded from the *ruleset*.
(Default: half of the interquartile range of the training set)

EXTRACT:

Training structures are fragmented into substructures according to the fragmentation parameters set in the options menu (**Options > Parameters...**). This operation can take several minutes. Then a *ruleset* is extracted and loaded in **QSARpy Predictor**.

A *ruleset* is composed of a set of *base structures* (the training structures) each of them associated to an activity value, and a set of *modulators* derived from the analysis of the structural partitioning of the training set, that are structural fragments associated to a fixed activity value variation (called *activity shift*).

VALIDATE:

The *ruleset* is validated against the molecular structures in the training set and the resulting statistics are prompted to the user.

At this point, you can:

- Modify the parameters (**Options > Parameters...**) and extract a new *ruleset*
- Modify the error threshold and extract a new *ruleset*
- Save the predictions on the training set in CSV file format from the file menu (**File > Save Predictions (Train)...**)

- Save the *ruleset* in a human-readable format from the file menu
(**File > Save Ruleset (human-readable)...**)
- Save the *ruleset* to use it in the next QSARpy sessions from the file menu
(**File > Save Ruleset...**)
- Save a training analysis of the extracted modulators from the file menu
(**File > Save Training Analysis...**)
- Hit **<Load a Test set>** and test new structures with the extracted *ruleset*
(→ *redirect to QSARpy Predictor*)

OR

- Start a new **QSARpy Extractor** session by loading a new dataset
(**File > Load Training set...**)

3. QSARpy Discovery

To start a new session load a *ruleset* from the file menu: **File > Load Model...**

SET MODEL TOLERANCE:

By setting an error tolerance threshold (that can be interpreted as the experimental error in the input data), it is possible to generalize the model in 3 automated steps:

Removing redundancies

When a modulator structure is the union of the structures of other smaller modulators, and the difference in the relative *activity shifts* is less than the threshold, then the bigger modulator is redundant. It is removed and the smaller and more generic modulator are used instead.

Placeholders

Modulators related to a very small *activity shift* (absolute value less than the threshold) are not really significant. They are kept just as '*placeholders*' with no *activity shift*: they are substructures known to not to alter the activity of the structure to which they are bonded.

Clustering

When more modulators share the same structure and they differ only for the connection bonds (the bonds with an asterisk in the SMILES of the fragment, e.g., 'C*' and 'C=*'), if their *activity shifts* are in a range of twice the threshold, they are clustered together in a single meta-modulator that matches with all the SMILES. The activity shift of such meta-modulator is an average of the single *activity shifts*.

For example: 'C*' and 'C=*' can be clustered in a meta-modulator 'C'.

GENERALIZE:

The generalized model is printed on the screen and automatically loaded into **QSARpy Predictor**.

Hit **<Reset>** to restore the original model

VALIDATE:

The generalized *ruleset* is validated against the molecular structures in the training set and the resulting statistics are prompted to the user. Accuracy can slightly differs from the original model, depending on the chosen threshold.

At this point, you can:

- Modify the model tolerance and get a new generalized *ruleset*
- Save the predictions on the training set in CSV file format from the file menu (**File > Save Predictions (Train)...**)
- Save the generalized *ruleset* in a human-readable format from the file menu (**File > Save Ruleset (human-readable)...**)
- Save the generalized *ruleset* to use it in the next QSARpy sessions from the file menu (**File > Save Ruleset...**)
- Hit **<Load Test set>** / **<Apply to Test set>** and test new structures with the generalized *ruleset*
(→ *redirect to QSARpy Predictor*)

OR

- Start a new **QSARpy Discovery** session by loading a new *ruleset*
(**File > Load Model...**)

4. QSARpy Predictor

To start a new session load a test set from the file menu: **File > Load Test set...**

See also “Load a dataset” for file format details (Chapter 1: START Page).

SET TEST SET HEADERS:

Specify the required field headers in the test set file:

- **ID:** CAS number or any other ID suitable to identify the molecular structure
- **SMILES** (CSV file format only)
- **Prediction Target:** the activity/property numerical value to be predicted

'ID' and 'Prediction Target' fields are optional. However, if the Prediction Target is not set it will not be possible to output prediction statistics.

Hit **<Load>** to load the molecular structures of the dataset as testing structures.

RULESET:

Load a *ruleset* saved in a previous session.

If a *ruleset* has been generated by **QSARpy Extractor** during the present session, such *ruleset* is already loaded and displayed.

PREDICT:

Testing structures are fragmented into substructures according to the fragmentation parameters set in the options menu (**Options > Parameters...**). This operation can take several minutes.

Then the *ruleset* is applied and for every structure S such that:

$$S = \text{Base Structure} \pm \text{Modulator 1} \pm \dots \pm \text{Modulator } n$$

“ \pm ” before a modulator means the structural addition or removal of the related substructural pattern

a prediction is assigned as a result of each contribution:

$$\text{Activity (predicted)} = \text{activity value}_{BS} \pm \text{activity shift}_{M1} \pm \dots \pm \text{activity shift}_{Mn}$$

If the *Detect outliers* option is checked (see the “Parameters” section), the structures are first preprocessed for similarity to outliers.

VALIDATE: (Only if Prediction Target is set)

The *ruleset* is validated against the molecular structures in the test set and the resulting statistics are prompted to the user.

At this point, you can:

- Load a new *ruleset* and test it against the test set from the file menu (**File > Load Ruleset...**)
- Save the predictions on the test set in CSV file format from the file menu (**File > Save Predictions (Test)...**)
- Start a new **QSARpy Predictor** session by loading a new dataset (**File > Load Test set...**)

4. Parameters

Fragmentation Algorithm parameters

These parameters are used by the QSARpy fragmentation algorithm to break down molecular structures into substructures. Modifying training parameters will affect the resulting model (the *ruleset*), modifying testing parameters will affect how the model is applied (i.e. how deep to search for an applicable rule).

Increasing the fragmenting parameters can improve the overall performance, but it could drastically affect the time and memory requirements. Increase them gradually.

FRAGMENT SIZE: (default = 28)

The maximum admitted number of atoms in a substructure.

FRAGMENTATION DEPTH LEVEL: (defaults training = 2, test = 3)

The number of iterations in the fragmentation algorithm.

The output of each iteration is the collection of the structural partitions achievable by breaking one bond in respect to the input structure. Every next iteration is applied to the output of the previous. Therefore, the first iteration produces the collection of the structural partitions made up of two fragments, the second iteration produces the collection of the structural partitions made up of three fragments, and so on. Example:

Input structure: ABCD

Iteration #1: A BCD ; AB CD ; ABC D

Iteration #2: A B CD , A BC D ; A B CD , AB C D ; A BC D , AB C D

...

HINTS:

While increasing testing parameters does not have any evident drawback beside time and memory cost, deep fragmenting the training set leads to a growth of the model complexity.

In particular, it is advisable to raise training fragmentation depth level above 2 iterations only if the results are not satisfying, and if the training set is not too big (>10000 compounds).

N.B. In order to fully apply the model keep testing parameters greater than or equal to training parameters.

New v1.1: Detect outliers

Outliers are identified in the training set (using the interquartile range rule) and included in the ruleset, but they are excluded from the learning process (i.e., no modulators are extracted from an outlier

structure). When a *ruleset* is applied, testing structures are preprocessed for similarity (*Tanimoto coefficient* = 1) with the outliers in that *ruleset*, in case of match is assigned a prediction by Similarity given by the median activity value of the similar structures.

5. Output formats

Unless otherwise specified, all output files are in CSV format and can be opened by any spreadsheet editor.

RULESET:

→ **File > Save Ruleset...**

This file is an internal format (.qsarpy) to store a *ruleset* and make it available to be loaded inside another QSARpy session. For convenience, the whole fragments hierarchy of the training set is also stored in the same file, so that it is not necessary to fragment it again in order to use the *ruleset*.

→ **File > Save Ruleset (human-readable)...**

This file is for user's convenience only; it cannot be loaded in a QSARpy session. It contains three sections: MODULATORS, BASE STRUCTURES and OUTLIERS.

MODULATORS

ID: the modulator's ID

SMILES: the SMILES

Activity SHIFT: the effect of the modulator on the activity value of a *base structure* once bonded to it

BASE STRUCTURES and OUTLIERS

ID: the structure ID field chosen by the user

SMILES: the SMILES structural line notation

Activity value: the activity value of the structure (as per training set)

PREDICTIONS:

→ **File > Save Predictions (Train)...** or **File > Save Predictions (Test)...**

ID: the structure ID field chosen by the user

SMILES: the SMILES structural line notation

Status: predicted/unpredicted, in brackets if they are predicted by similarity to Outliers or by identity with a structure in the training set

Structural factorization (SMILES): the structural factorization containing the SMILES of the *base structure* and the modulators, sign + if the modulator's substructural pattern has to be "added" to the *base structure*, sign - if it has to be removed

Structural factorization (IDs): same as above but with IDs instead of SMILES

Modulation sequence: how the predicted activity value changes from its initial value (the *base*

structure's activity value) to the final value due to the subsequent contribution of each modulator

Prediction: the final predicted activity value

The following fields are present only if the user has provided a Prediction Target:

Target: the actual activity value

ERROR: the prediction error

TRAINING ANALYSIS:

→ ***File > Save Training Analysis...***

This file is the rationale of the *ruleset*. It contains, for each modulator, all the observations (from the training set) from where the *activity shift* value has been derived.

ID: the modulator's ID

SMILES: the SMILES

TRAINING ref. ID: the reference ID in the training set of the structure in which a modulation has been observed

Structural factorization (IDs): the training structure partitioning that highlight the modulation: ID of the *base structure* and ID of the modulator(s)

Structural factorization (SMILES): same as above but with SMILES instead of ID

Equation: the linear equation from which can be calculated the actual *activity shift* value of the modulator

INPUT: the activity value before the modulation

OUTPUT: the activity value after the modulation

DELTA: OUTPUT - INPUT

Activity shift (median): the final activity shift of a modulator is defined as:

$median(outputs) - median(inputs)$, where *outputs* and *inputs* are respectively the set of every OUTPUT and INPUT values for the modulator under analysis

SESSION LOG:

Hitting the **< Save LOG >** button in the bottom right of a QSARpy tab will save a text file (.txt) with the content of the output window of the present session. As a suggestion, consider to always store a saved model together with the log of the session that generated it, which contains all the information to reproduce the model.