Manual of
SPCI (structural and physico-chemical interpretation) open-source software
version 0.1.3

Version (date)	Changes and comments
0.1.0 (02.02.2015)	Changes from alpha version: 1. More precise default SMARTS was added. 2. Cross-validation calculation was sped up and intermediate predictions are saved in text files. 3. Compounds, which cause errors in calculation of atomic properties with Chemaxon cxcalc tool, are excluded from further modeling. 4. Intermediate results of fragments contributions calculation are saved to a text file.
0.1.1 (07.02.2015)	1. Fixed errors in text output of intermediate results of fragments contributions calculation. 2. Fixed error in loading of file with descriptors on 32-bit platforms.
0.1.2 (13.05.2015)	Fixed error in loading of a file with descriptors on 32-bit platforms.
0.1.3 (22.07.2015)	Added two automatic fragmentation schemes: detection of i) all rings and ii) Murcko frameworks.
0.1.4 (21.01.2016)	Added automatic fragmentation scheme based on SMARTS to cleave bonds. Changed routine of calculation of simplexes with hydrogen bonding labels. Simplexes with no labels A (acceptor), D (donor) or AD (acceptor and donor) are discarded. Added support of multiple modeling properties in a single sdf-file (if property is missing put NA and such compounds will be discarded during modeling). Default format of descriptors files is svm now (more affective to store sparse data). GUI changes: - 10 last paths of loaded sdf-files are stored and can be easily loaded again - user may specify the number of cores to use for model building (default value is max_cores - 1) Backward compatibility is broken, old projects cannot be opened with this new version due to changes in project management.

Content

Overview

The SPCI software was designed for (semi)automatic extraction of structural features and their contributions to an investigated property from chemical datasets. It's a bunch of Python3 scripts. In order to simplify the usage a GUI was developed. It has a limited number of options but suits well for most needs.

Concept and citations

The idea of structural and physico-chemical interpretation along with comparison with MMP approach is disclosed in the following references. Please cite them if you use this software.

1) Polishchuk, P. G.; Kuz'min, V. E.; Artemenko, A. G.; Muratov, E. N. Universal Approach for Structural Interpretation of QSAR/QSPR Models. *Molecular Informatics* **2013,** *32*, 843-853.

2) Polishchuk, Tinkov, Khristova, Ognichenko, Kosinskaya, Varnek, Kuz'min Structural and physico-chemical interpretation of QSAR models, **2016** (in preparation).

Structural *(doesn't require Chemaxon)* and physico-chemical interpretation *(Chemaxon required)*

Structural interpretation returns only overall fragments contributions while physico-chemical interpretation can additionally estimate contributions of some physico-chemical factors (electrostatic, hydrophobic, hydrogen bonding and dispersive terms). Installed Chemaxon is required for physico-chemical interpretation.

If you have a license for Chemaxon check whether it is installed and PATH variable is correctly configured (to launch `standardize` and `cxcalc` from command line). If standardize or cxcalc command are not recognized from your command line, add JChem bin folder to PATH variable.

Installation and launch.

There are some dependencies which should be installed:

	Tested environments	
	Windows	Linux
Python	>= 3.2	3.4
matplotlib	1.4.3	1.3.1
numpy	1.9.2	1.10.4
scipy	0.15.1	0.16.1
scikit-learn	0.16.1	0.17
indigo toolkit	1.1.12	1.2.1

Windows

You may install all Python dependencies by yourself and download the latest version of the application from github in order to install and upgrade it (see Linux installation section for some details).

To make life easier for Windows users WinPython distributions (32 and 64-bit) were prepared and zipped. They contain all required dependencies for immediate start.

Step 1. Visit the page http://qsar4u.com/pages/sirms_qsar.php, download a desired WinPython distribution and unzip the archive into a folder.

Step 2. Download spci.zip and unpack into the same folder where WinPython was unpacked. You should obtain an installation folder with three items: `spci` and `WinPython` folders and `start-spci.bat`.

Program_installation_folder

```
|-spci/
|-WinPython/
|-start-spci.bat
```

To launch the application choose `start-spci.bat`.

Update:

To update the application just remove `spci` folder from the installation folder and unzip newly downloaded `spci.zip` in it.

Linux (Ubuntu):

Python 3.4 in Ubuntu is shipped with almost all required packages.

I would recommend to upgrade `numpy` and `scipy` packages:

```
sudo pip3 install -U numpy
```

```
sudo pip3 install -U scipy
```

Run the following command to install missing libraries:

```
sudo apt-get install libatlas-dev libatlas3gf-base
```

Then install `scikit-learn`:

```
sudo pip3 install scikit-learn
```

And make sure that atlas is used to provide the implementation of the BLAS and LAPACK linear algebra routines:

```
sudo update-alternatives --set libblas.so.3 \
    /usr/lib/atlas-base/atlas/libblas.so.3
```

```
sudo update-alternatives --set liblapack.so.3 \
    /usr/lib/atlas-base/atlas/liblapack.so.3
```

Download and install Indigo toolkit -
http://lifescience.opensource.epam.com/download/indigo/index.html

Then clone the repo:

```
git clone https://github.com/DrrDom/spci
```

Go into the created `spci` dir and init and update submodules:

```
git submodule init
```

```
git submodule update
```

To start the application call `python3 spci.py` from `spci` dir.

Update:

Update spci repository

```
git pull origin master
```

Update submodules if necessary

```
git submodule update
```

Workflow:

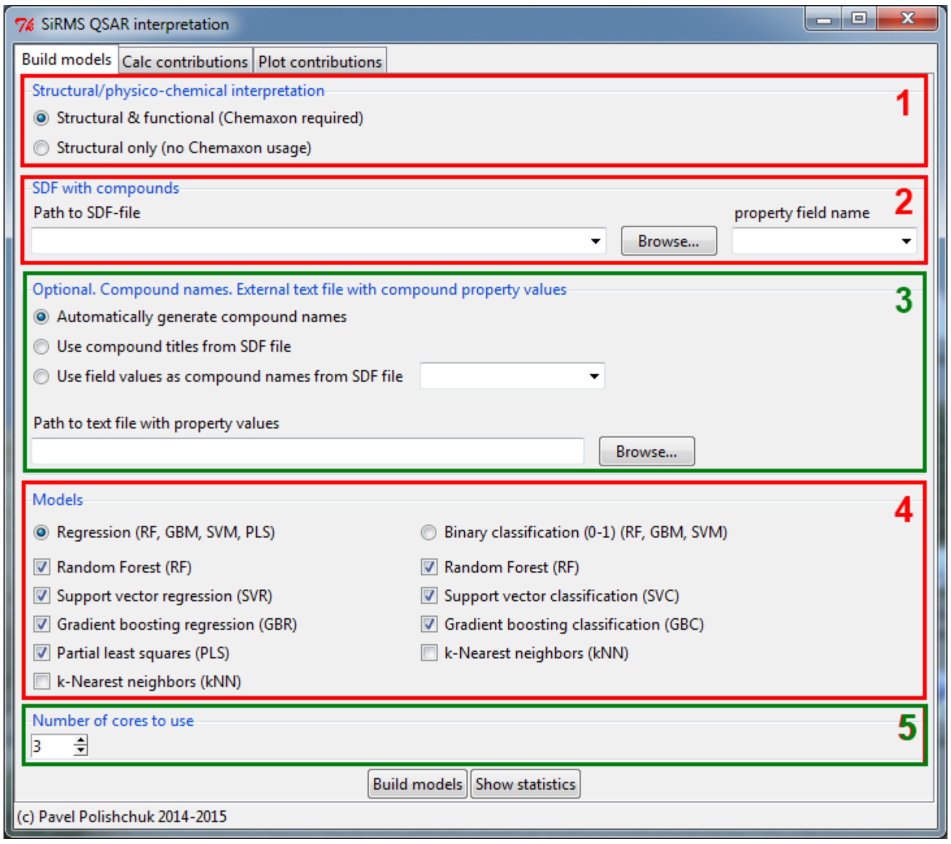
Step 0. Data preparation and project start.

For analysis a single sdf file is required, which contains compounds' structures and corresponding property value as a field in sdf file. Input sdf file my contain several modeling properties. The structures should be checked and standardized before run analysis (this is absolutely required in the case of only structural interpretation model, otherwise results can be distorted because standardization influence calculated descriptors).

As an alternative, you may prepare separate text file with compounds' property values (file must have a header and two columns compound name and property value separated by tab).

Place the sdf file in a separate folder which will be a project folder.

Step 1. Build models.


SiRMS QSAR interpretation

Build models | Calc contributions | Plot contributions

1 Structural/physico-chemical interpretation
- [✓ Structural & functional (Chemaxon required)]
- Structural only (no Chemaxon usage)

2 SDF with compounds
- Path to SDF-file:
- property field name:

3 Optional. Compound names. External text file with compound property values
- [✓ Automatically generate compound names]
- Use compound titles from SDF file
- Use field values as compound names from SDF file
- Path to text file with property values:

4 Models
- [✓ Regression (RF, GBM, SVM, PLS)]
  - [✓ Random Forest (RF)]
  - [✓ Support vector regression (SVR)]
  - [✓ Gradient boosting regression (GBR)]
  - [✓ Partial least squares (PLS)]
  - k-Nearest neighbors (kNN)
- Binary classification (0-1) (RF, GBM, SVM)
  - [✓ Random Forest (RF)]
  - [✓ Support vector classification (SVC)]
  - [✓ Gradient boosting classification (GBC)]
  - k-Nearest neighbors (kNN)

5 Number of cores to use: 3

*Red fields are required, green ones are optional.

1. Choose the type of analysis to perform.
2. Specify path to sdf file with compounds and choose the name of the field containing property values. Paths to 10 last opened sdf files are stored.
3. Optional. It is required if you want to save original compounds names or if property values are supplied in a separate file.
4. Choose the desired type of models. All listed models will be build with optimal parameters selected by grid search in 5-fold cross validation procedure.
5. Optional. The number of cores to be used during models building. By default all cores minus 1 will be used.

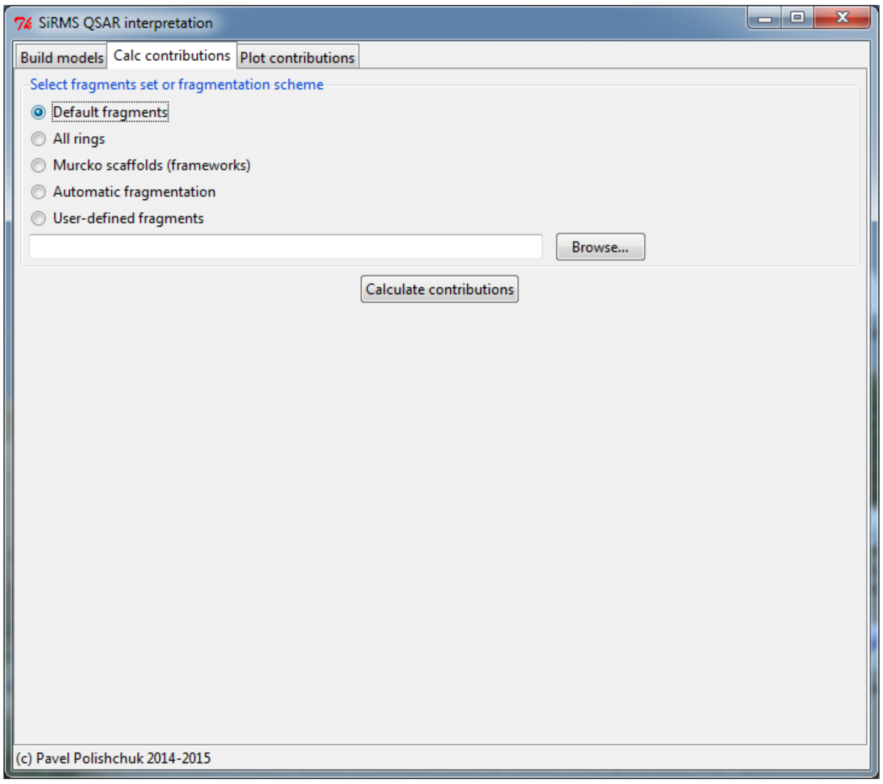
IMPORTANT:

To open an earlier created project you should choose the sdf file and property field name.

Step 2. Calculation of the fragments' contributions.

Contributions will be calculated only for those models specified on the previous tab.

If models had reliable predictive ability estimated by 5-fold cross-validation (Show statistics button) one may calculate fragments contributons.


SiRMS QSAR interpretation

Build models | Calc contributions | Plot contributions

Select fragments set or fragmentation scheme

- [✓ Default fragments]
- All rings
- Murcko scaffolds (frameworks)
- Automatic fragmentation
- User-defined fragments

Browse...

Calculate contributions

(c) Pavel Polishchuk 2014-2015

default fragments – common functional groups and rings (specified in default.smarts file in program folder);

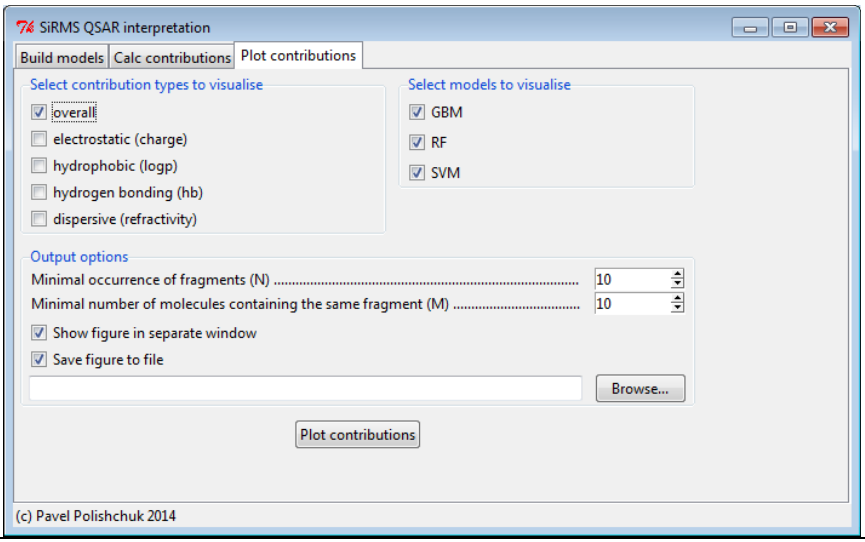
all rings – automatically detect all single and fused rings;

Murcko frameworks – automatically detect Murcko frameworks;

automatic fragmentation – split molecules on maximum three parts on bonds match SMARTS: [#6+0;!$(*=,#[!#6])]!@!=!#[*]

user-defined fragments represented in SMILES or SMARTS notation, look at default.smarts as a file format example (Note: SDF usage was not tested).

Step 3. Plot contributions and save plot in png file.


SiRMS QSAR interpretation

Build models | Calc contributions | Plot contributions

Select contribution types to visualise

- [✓] overall
- [ ] electrostatic (charge)
- [ ] hydrophobic (logp)
- [ ] hydrogen bonding (hb)
- [ ] dispersive (refractivity)

Select models to visualise

- [✓] GBM
- [✓] RF
- [✓] SVM

Output options

Minimal occurrence of fragments (N) 10

Minimal number of molecules containing the same fragment (M) 10

- [✓] Show figure in separate window
- [✓] Save figure to file

Browse...

Plot contributions

(c) Pavel Polishchuk 2014

After project was opened all available models will be listed in order to choose them for visualization.

Overall contribution (structural interpretation).
Other types is used for physico-chemical interpretation (to show the contribution of separate physico-chemical factors).

Alternative step 3.

You may create alternative visualization in order to customize plot output:

1) by yourself (parsing *_frag_contribution.txt files with calculated contributions located inside the modeling property folders and applying any of available tools);

2) by using the developed web-base tool which is suitable for visualization of relatively small set of fragments (number of columns in *_frag_contribution.txt should be less than 100k) otherwise it will be painful;

Full version - http://158.194.101.252:3838/spci-vis/

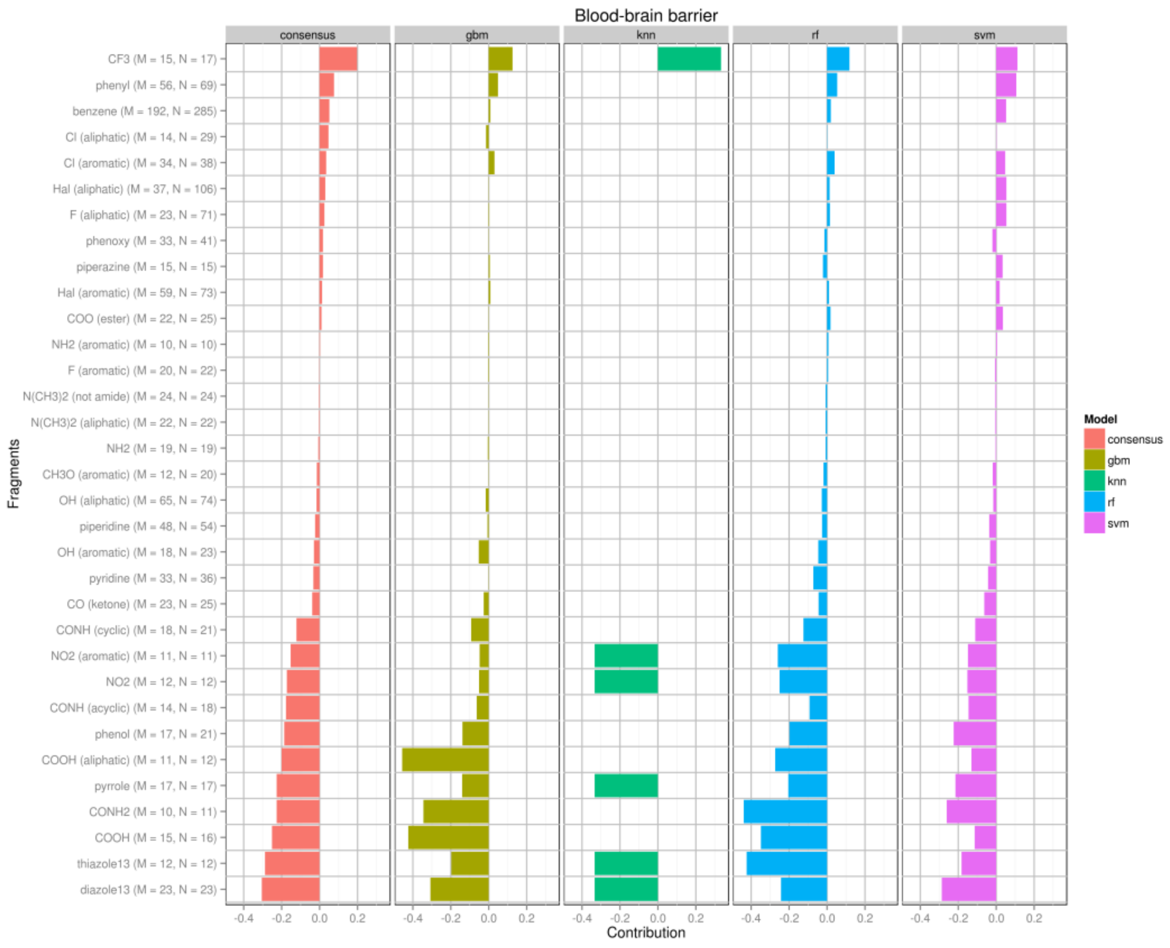
Demo version to just play with pre-loaded data - http://158.194.101.252:3838/spci-vis-demo/

3) by using `rspci` R package on your local machine. This package contains functions to facilitate data reading, modification, filtering and plot. It has more options that the web-based tool and recommended for advanced usage by R users.

To install `rspci` package call from R console:

```
devtools::install_github("DrrDom/rspci")
```

Below is an example of contributions plot:


Blood-brain barrier
consensus
gbm
knn
rf
svm
CF3 (M = 15, N = 17)
phenyl (M = 56, N = 69)
benzene (M = 192, N = 285)
Cl (aliphatic) (M = 14, N = 29)
Cl (aromatic) (M = 34, N = 38)
Hal (aliphatic) (M = 37, N = 106)
F (aliphatic) (M = 23, N = 71)
phenoxy (M = 33, N = 41)
piperazine (M = 15, N = 15)
Hal (aromatic) (M = 59, N = 73)
COO (ester) (M = 22, N = 25)
NH2 (aromatic) (M = 10, N = 10)
F (aromatic) (M = 20, N = 22)
N(CH3)2 (not amide) (M = 24, N = 24)
N(CH3)2 (aliphatic) (M = 22, N = 22)
NH2 (M = 19, N = 19)
CH3O (aromatic) (M = 12, N = 20)
OH (aliphatic) (M = 65, N = 74)
piperidine (M = 48, N = 54)
OH (aromatic) (M = 18, N = 23)
pyridine (M = 33, N = 36)
CO (ketone) (M = 23, N = 25)
CONH (cyclic) (M = 18, N = 21)
NO2 (aromatic) (M = 11, N = 11)
NO2 (M = 12, N = 12)
CONH (acyclic) (M = 14, N = 18)
phenol (M = 17, N = 21)
COOH (aliphatic) (M = 11, N = 12)
pyrrole (M = 17, N = 17)
CONH2 (M = 10, N = 11)
COOH (M = 15, N = 16)
thiazole13 (M = 12, N = 12)
diazole13 (M = 23, N = 23)
Fragments
Contribution
Model
consensus
gbm
knn
rf
svm