Tutorial. From QSAR modeling to outliers detection

Introduction

Goal: Illustrate a detection of outlier protocol based on a QSAR modeling workflow.

Software: WEKA, ISIDA/ModelAnalyzerR, InstantJChem
Data: A sample of the IUPHAR [1] database dedicated to the serotonin receptor 5-HT_{2B} (IUPHAR_5HT2B.sdf), a set of publications (Publications.zip).

The dataset contains ligands of the human serotonin receptor 5-HT_{2B}, a member of the 5-HT2 family involved in morphogenesis and anxiety [2]. But this receptor became an anti-target, since it was suspected to have a role in the development of valvular heart disease [3-5]. This discovery led to highly publicized drug withdrawal from the market, first in the Fenfluoramine/Phentermine case [6] and more recently in the benfluorex case [7].

The data for this tutorial were collected from the IUPHAR/BPS [8]. It is composed of 88 compounds together with their affinity to 5-HT_{2B}, their role either as agonist or antagonist, and the PubMed [9] references to the articles describing these information. If several values were reported for a given ligand, the retained value is the median of the collected values. All values are reported for human receptor, if available. Otherwise, the values for the rat receptor are provided (species is always mentioned together with the affinity value). All the articles used to collect this dataset are provided into the archive Publications.zip. The files of the articles are named according to their PubMed ID.
Compound structures and related data fields are in file IUPHAR_5HT2B.sdf. The SD fields `affinity`, `type` and `pubmed_id` contain the pKi value, the agonist or antagonist label and the list of bibliographic references respectively. The SD field `CdId` contains a unique integer to facilitate reference to a given compound and the `ligand` field contains the common name of the compound.
The regression problem consists in estimating ligand affinity (expressed as pK_i = negative log of the complex instability constant expressed in mol/l) to 5-HT_{2B}, as a function of the ligand structure.

Theoretical background.

The tutorial uses Gaussian Processes to build models [10, 11]. The general idea is to describe the dataset as a multivariate Gaussian probability distribution. The vector of target affinities, Y, is understood as a sample of a multivariate normal distribution resulting from the molecular descriptors, and a noise σ. The molecular descriptors are contributing to the definition of the distribution by estimating the covariance of the distribution. Technically, the covariance is identified to a kernel function of the training set instances: $\Sigma_{ij} = k(x_i, x_j)$.

The method depends of several choices: the level of noise and the kernel function that is also parameterized. The noise level is therefore related to the confidence in the values of affinities, while the kernel shall be chosen based on domain knowledge about the molecular descriptors in use.

Although it is not needed in the tutorial, the Gaussian Processes are often embedded into a global Bayesian reasoning. In this case, the parameters (noise, kernel parameters) are themselves sampled to maximize the marginal likelihood using dedicated algorithms such as Markov Chain Monte Carlo [12]. These approaches, combined to the numerical complexity of the linear algebra needed to solve the Gaussian Processes problems, can lead to rather costly calculations for datasets of thousands of instances.

The tutorial adds some focus to the detection of outliers. The strategy proposed here consists into developing the best "non-over-fitted" model on the dataset. To avoid over-fitting, a rigorous external validation is recommended. Then, if the model cannot fit some instances, they are potential outliers. Once the outliers are removed, model building and outlier analysis is iterated. The process is ended when no more outliers are found. This is a sequential inward approach [13-15]. An outward approach would consist in adding non-outlier data to the dataset in a sequential procedure. It is possible to search for ensembles of outliers, or to do the stepping by focusing on the single most outstanding outlier at every stage.

Irrespective of the algorithm, there are several important aspects to be kept in mind during outlier analysis.

- First, an outlier is not the result of a numerical procedure: an algorithm fails to fit a data point, but this may be called an outlier only if the anomalous value can be discarded for a reason: a potential measure problem, an unexpected event during data acquisition, sabotage… Without a cause, an anomalous point is not an outlier and cannot be discarded – on the contrary, it may simply be an indication of the failure of the modeling strategy.
- Second, a data point can be anomalous only relative to some *a priori* knowledge. In this tutorial, we propose for instance to use the Grubbs algorithm [16] which is very representative: an anomalous point is an extreme value compared to a normally distributed sample.

In this tutorial, we shall focus on the distribution of the residuals of a regression model. They are supposed to follow a Normal probability distribution. The n residues of values r_i are searched for the largest value r_n and the smallest value r_1. It turns out that the quantity $G_n = \frac{r_n - \langle r \rangle}{s}$ and $G_1 = \frac{\langle r \rangle - r_1}{s}$, where s and $\langle r \rangle$ are the standard deviation and mean of the residuals, follow a studentized extreme deviation statistics [17]. The Grubbs test, consists in comparing these decision variables, to the critical value $G_c = \frac{n-1}{\sqrt{n}} \sqrt{\frac{t^2}{n-2+t^2}}$, where t is the ${}^{\alpha}/_{2n}$ fractile of a student distribution of $n-2$ degrees of freedom, with a risk α. If a decision variable is larger than the critical value, the extreme value shall be considered as anomalous. By convention, if for a data value the test is positive up to a risk α of 1%, it is considered anomalous whereas with a risk of 5% it is only suspicious.

This setup is consistent as long as the hypothesis that the residuals are distributed according to the centered normal law is legitimate. This is not the always the case, as for instance for the residuals of a fit from of an ϵ-SVM model.

Step by step instructions

Exercise 1 (Optional):

In this first part of the tutorial, the IUPHAR_5HT2B.sdf file is loaded into *InstantJChem*. It is essential to work with a database, in order to include or exclude easily some instances and to easily check all available information on a given instance.

Instructions	***Comments***
• Start *InstantJChem*. • Click on the main *File->New Project…* or alternatively click on the icon indicated on Figure 1.	Start the software *InstantJChem* (Figure 1). Start a new project to open the *New Project* wizard.

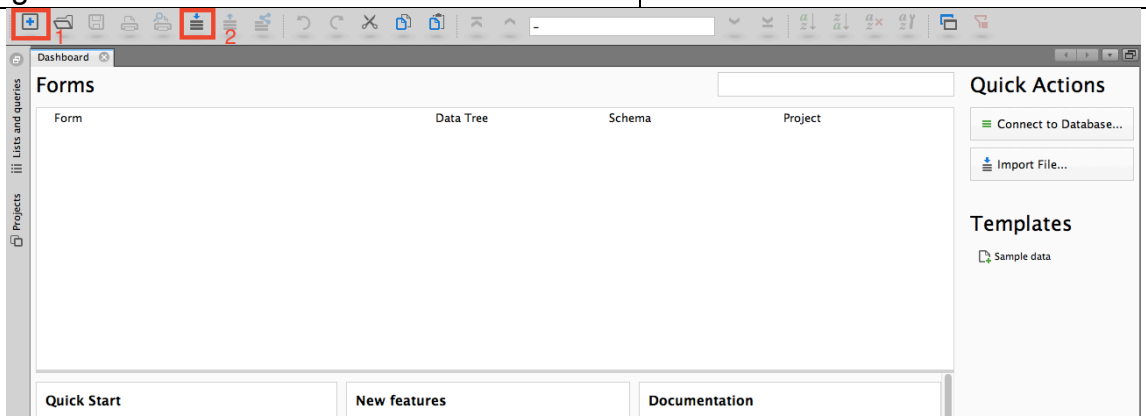

Figure 1: *The InstantJChem software starts with a Dashboard summarizing user's projects. To start a new project, click on the framed icon 1 in red in this picture. To import a dataset click the framed icon 2.*

• Select the *IJC Project (with local database)* option, and then click on *Next* (Figure 2). • In the next step, name your project as *5-HT2B* in the field *Project Name*. • Choose a proper folder in the field *Project Location* (Figure 3). A good choice can be the folder containing the tutorial files. • The field *Project folder* is adequately set automatically. • Click the *Finish* button.	These operations create a new directory named *5-HT2B* that will contain the database. Yet, the database contains no data so far. The *5-HT2B* directory will contain everything that will be needed to manage a local database.

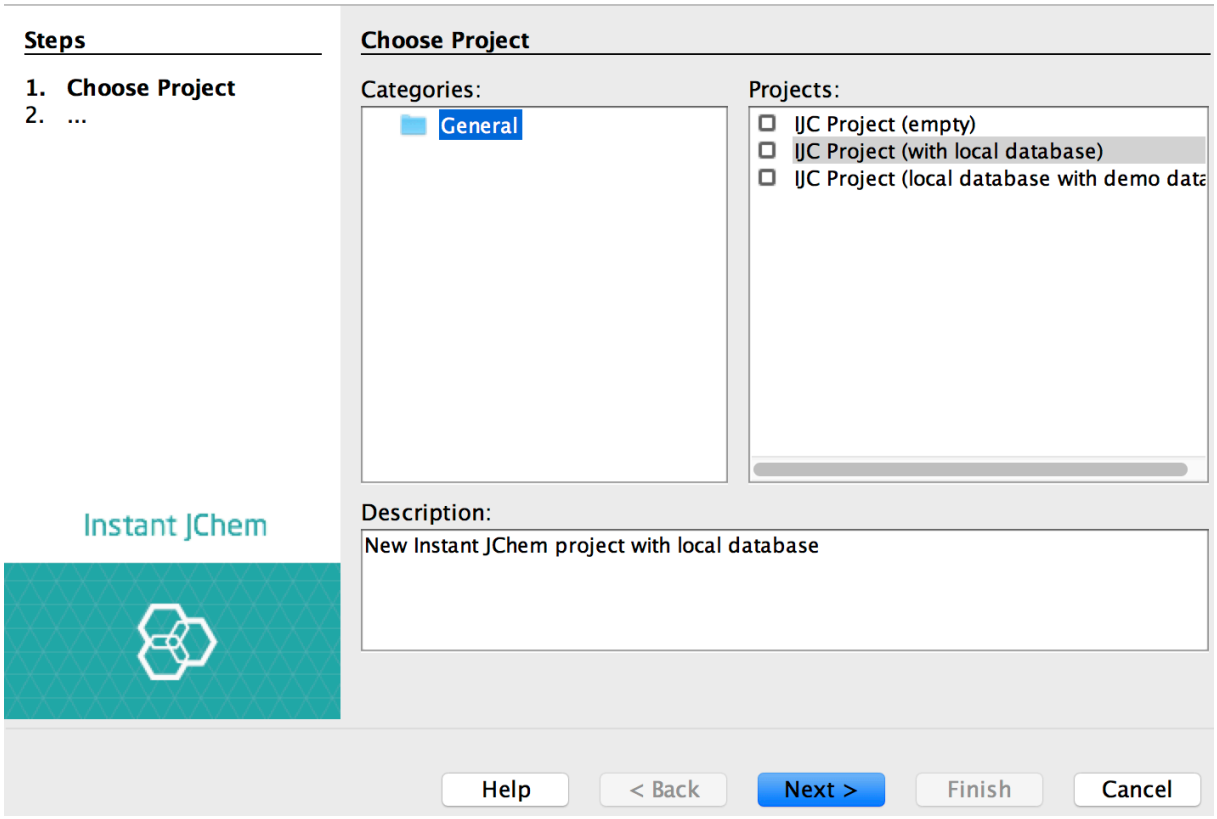

Figure 2: *Project creation wizard. Chose the second option that creates a project and setup a local database in one step.*

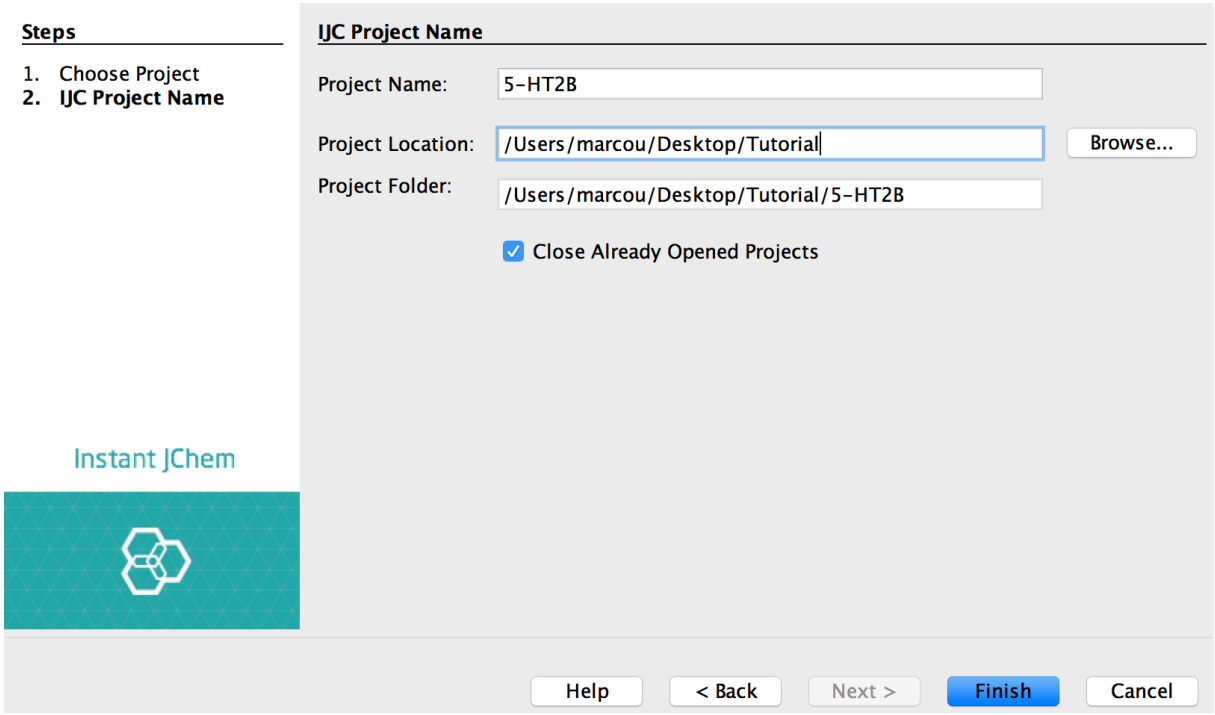

Figure 3 : *Second step of the project creation wizard. Once a project name and location is set, the project folder is set automatically.*

• Click on the main *File->Import File…* or alternatively click on the  icon indicated (see Figure 1). • Choose the file `IUPHAR_5HT2B.sdf` then click on the *Next* button (Figure 4). • Then check the frame labeled *Field in file*. Click on the item *CdId* and then on the button *Add*. • Set the *Display name* of this field as *Original CdId* (Figure 5). • Click the button *Next*. • At the end of the loading process, click the button *Finish*.	The wizard guides the user through the process of importing the chemical structures with their additional information into the *5-HT2B* database. During the process, the original *CdId* field from the file is not automatically imported because it seems to be a duplicate of the default *CdId* field generated by *InstantJChem* during import. It is therefore necessary to force the import and to rename it as *Original CdId*. The name of the database field is automatically updated from the display name.

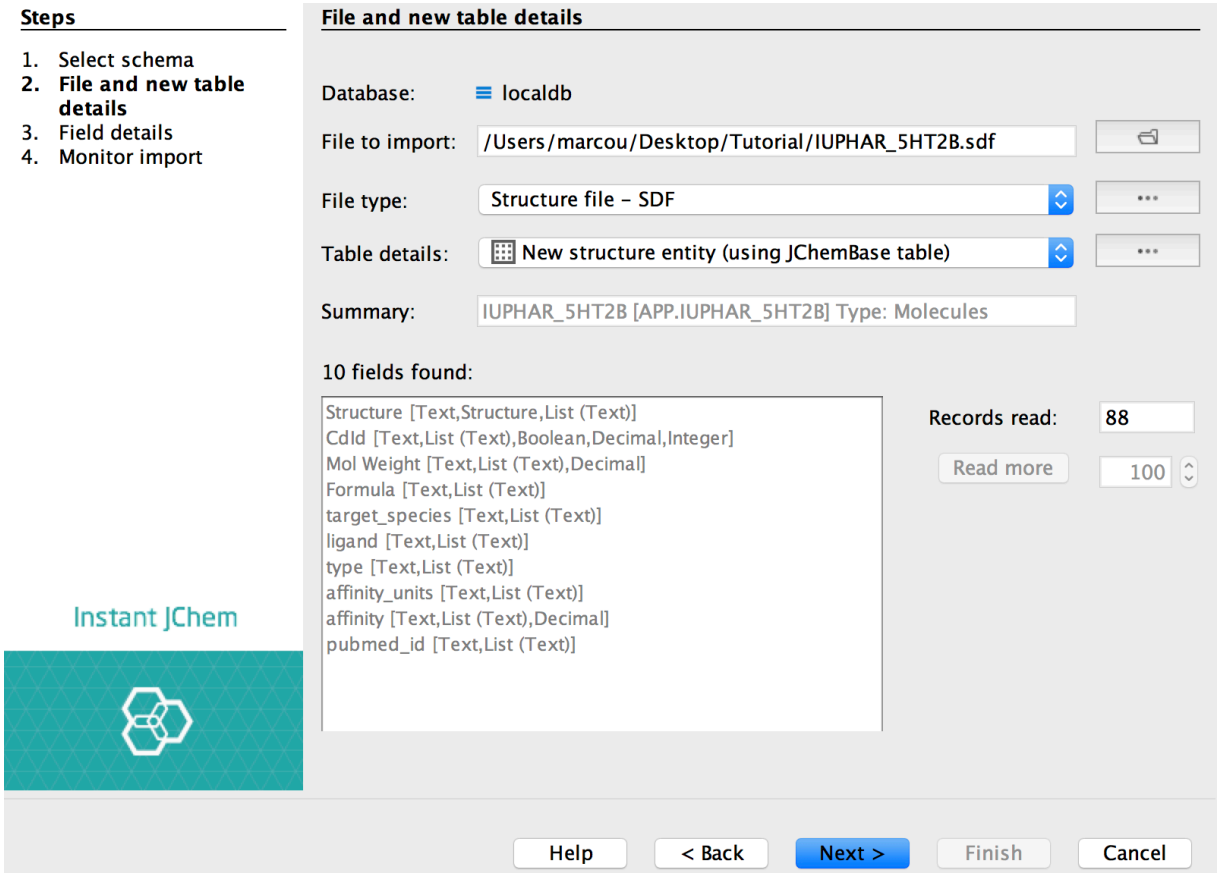

Figure 4: *File and table details of the import wizard.*

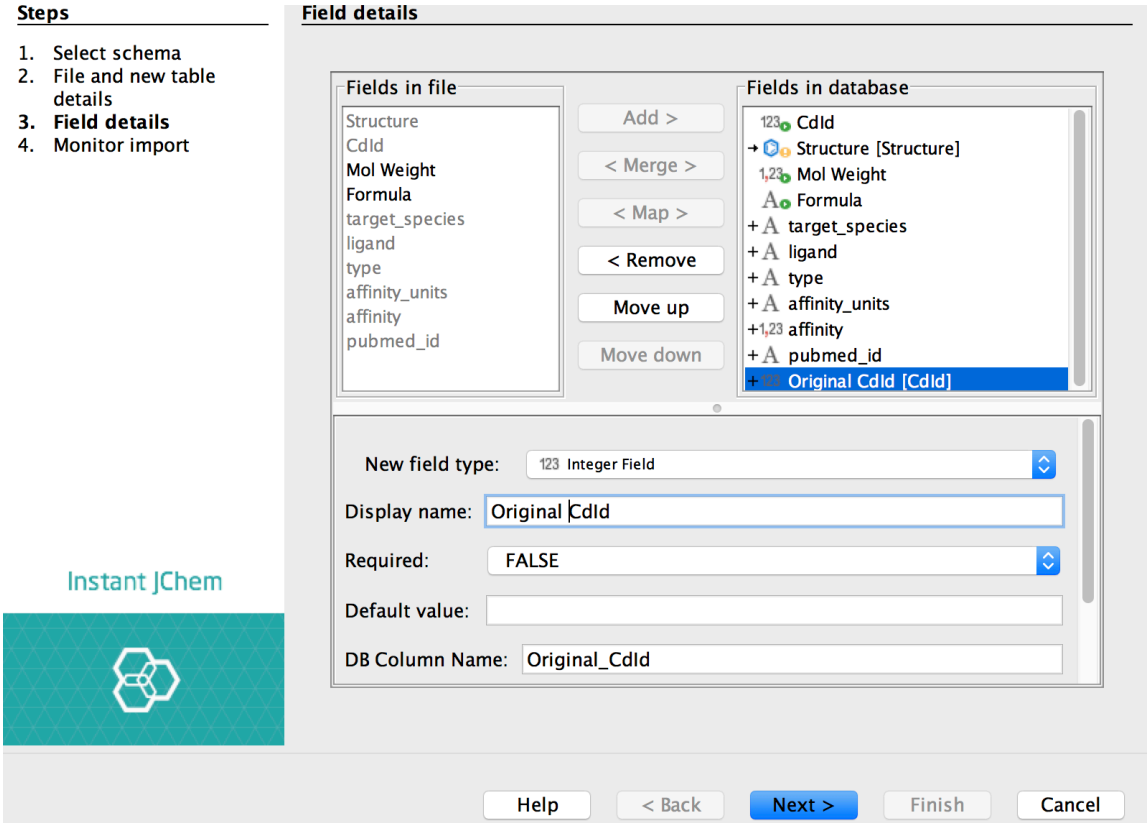

Figure 5: *Field details interface. Be sure to import the CdId from the input file, after renaming it "Original CdId".*

• In the grid view of the *5-HT2B* database, click the button ··· located in the top left corner and select the *Open Column Manager* option (Figure 6). • Select the items *CdId*, *MolWeight* and *Formula* from the right hand frame then click the Remove button (Figure 7).	The automatically generated fields are disturbing for the management of the database in the present case. Therefore, it is advised to hide them. The database is now ready to use (Figure 8).

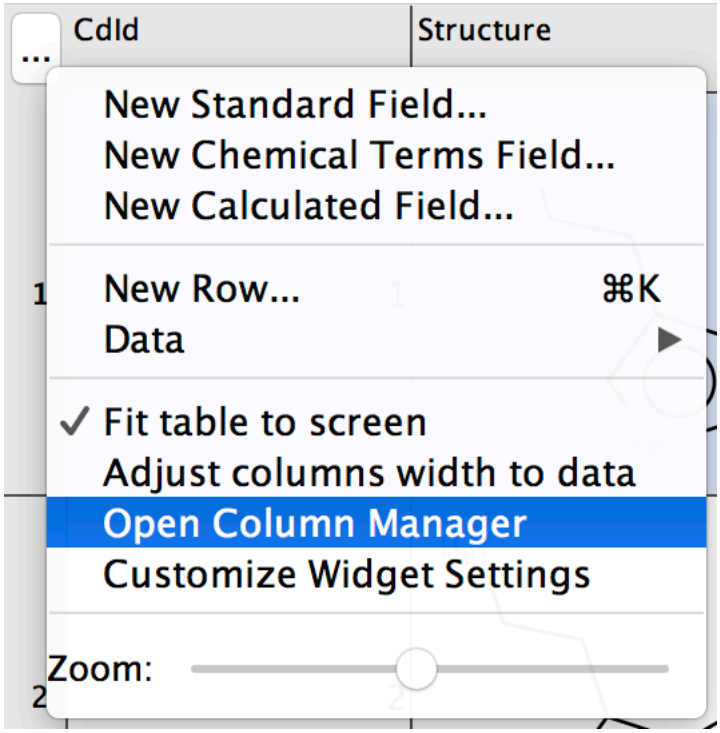

Figure 6: *The widget configuration menu of the grid.*

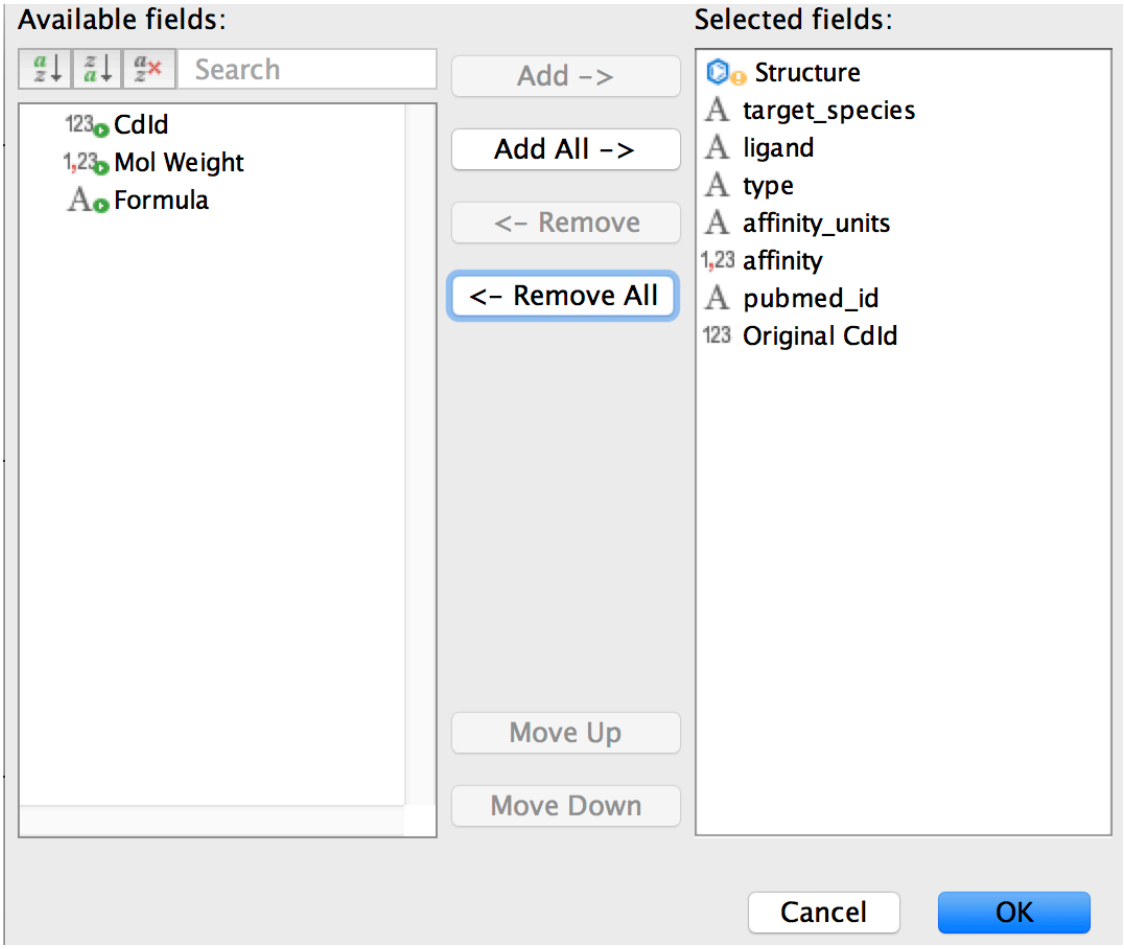

Figure 7: *The column manager interface. Remove confusing columns: CdId, MolWeight and Formula.*

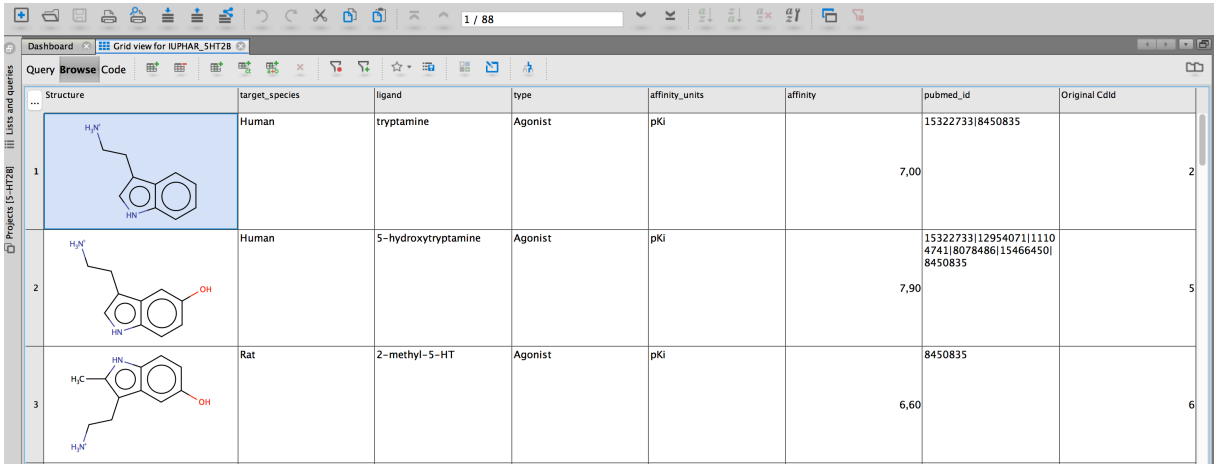

Figure 8: *State of the InstantJChem interface, once the 5-HT2B data has been loaded.*

Exercise 2:

In the second part, the `IUPHAR-5HT2B.sdf` file is used to generate an external cross-validation framework.

• Start the *ExtCV* software (Figure 9). • Use the interface (1, in the figure) to select the file `IUPHAR-5HT2B.sdf`. • Choose a cross-validation experiment in the menu (2) indicated on the figure. • Edit the boxes (3) to set the number of folds to *N*=4 and the number of repetition of the experiment to *k*=1. • Uncheck the *Create separated folders* box. • Click the button *Run!*.	The tool will create new folders called `CV`. It contains `trainIteriFoldj.sdf` and `testIteriFoldj.sdf` files, where `i` and `j` refer to a particular iteration and fold respectively. A test set corresponds to the train set with the same values for `i` and `j`. During the procedure, the training set and test sets are isolated from each other. The calculation of the molecular descriptors being separated train and test sets, it is not possible to transfer information from the test set to the train inadvertently. Beside, it forces the modeler to produce a complete chain to apply his models to the test data.

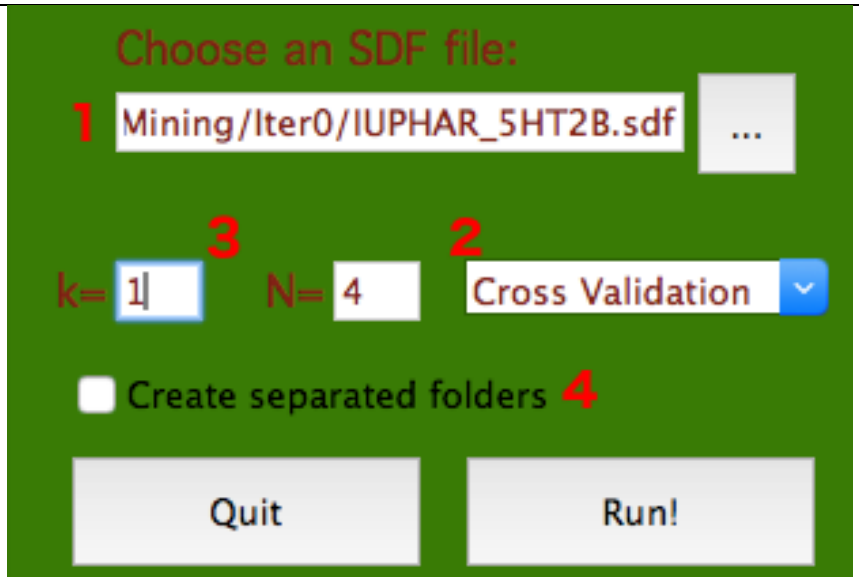

Figure 9: *Interface of the ExtCV software. The edit field and the button to its right (1) is used to select an SDF file, the combobox interface (2) is used to choose a validation protocole and the numeric text edit (3) are used to set the parameters of the validation (N iterations and k repetitions). If the box (4) is checked, a directory is dedicated to each fold and iteration; if not a common directory containing all train and test sets is created.*

Exercise 3:

The aim of the third part is to compute ISIDA Substructural Fragement Descriptors. The generation of these descriptors is dependent of the chemical structure files processed. Therefore, attention must be paid to produce the molecular descriptors in a consistent way between the training and test datasets.

- Start the *xFragmentor* software (Figure 10).
- Click the button *Add SDF* to add all the `train*.sdf` files located in the directories `CV`.
- Add also the file `IUPHAR_5HT2B.sdf`.
- Click the button *Add fragment* to add to the fragmentation algorithm, a simple atom count.
- Choose the *IIR* fragment topology from the *Select a topology* combo box.
- Set the *Min length* to 2 and the *Max length* to 4.
- In the combo box *Select a coloration scheme for atoms*, select the option *A*.
- In the combo box *Select a coloration scheme for bonds*, select the option *B*.
- Tick the box *Use formal charge*.
- Click the button *Add fragment*.
- Type the word “affinity” into the *SDF field of interest* text edit.
- Click the button *Run!*.

During this step, all the files `train.sdf` are analyzed and ISIDA substructural molecular fragment (SMF) descriptors are computed.

More specifically, these descriptors are counting the number of atoms of each type (fragments of type *E*) and the number of atom centered fragments composed of paths of uniform length, the length varying between 2 and 4 atoms. The atom types and bond types are recorded into the fragment and the formal charges, if any, are annotated (fragments of type *IIAB_R(2-4)_FC*).

In the folder containing a target SDF, the software creates several files:

	• An XML file recording the state of the software when it generated the ISIDA SMF descriptors. • A HDR file, containing the list of molecular fragment hashed from the chemical structures. • An SVM file that is a sparse storage of the molecular descriptor values. • An ARFF file that condense in one file the information of the HDR and SVM file, and can be used in the Weka data mining software. During the procedure, the values found in the field named *affinity* into the SDF file are reported into the molecular descriptor files. It the value is missing it is replaced by a question mark ("?"). This field is considered as the target property of the QSAR.
• Click the button *Save XML*. • Save configuration as `train_E_IIAB_R(2-4)_FC.xml`.	Save the current configuration of the software. It can be reloaded, so that exactly the same fragmentation can be reproduced later.
• Select all lines in the frame *List of SDF* files to process and click the button *Remove SDF*. • Click the *Add SDF* button and select each `test.sdf` file in the folders CV. • Tick the box *Use predefined fragments*. • Tick the box *Use only those fragments*. • Leave the text box to its default value (`Base name of header files…`). • You can check that there are no missing header files by clicking on the button *Check*. • Click the button *Run!*.	During this step, the ISIDA SMF descriptors are computed on the `test.sdf` files, using the same algorithm as the `train.sdf` files, and the same molecular fragment dictionary (the header files). By default, if the SDF file names contain the substring "`test`", the dictionary of fragments is searched to the corresponding `.hdr` file which base name contains the substring "train" instead. If new fragments are discovered in the process (and subsequently ignored), a "!" will appear in the log of

	the interface, near the index of the chemical structure. Therefore, the `CVtest*` files previously generated are ready to be used as external test sets for models build and optimized using the `CVtrain.*` files only.

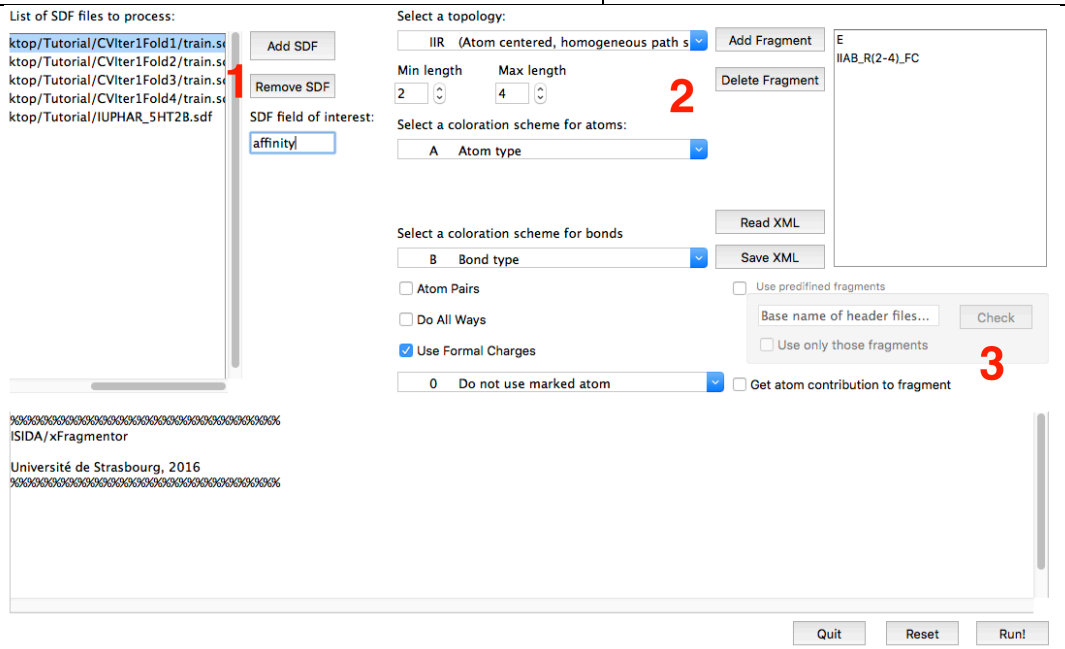

Figure 10: *Interface of the xFragmentor software. The zone (1) is dedicated to define the list of SD files to compute molecular descriptors. The zone (2) is used to setup and combine fragmentation algorithms. The zone (3) is used to generate molecular descriptors using predefined fragments.*

Exercise 4:

In the third part, Gaussian Process models are optimized on the training sets. Then, the model is applied to the corresponding test set.

• Open the *Weka* software.
• Choose the *Experimenter* tool (Figure 11).
• Click the button *New.*
• Type "ExtCVtrain.arff" into the *Results destination* text edit. You can chose the location of this file in your working directory, `Iter0` for instance.
• Into the *Experiment Type* frame:
  ○ Set the menu to *Cross-validation*
  ○ Set the *Number of folds* to 4
  ○ Select the *Regression* radio button
• In *Iteration Control* set the *Number of Iteration* to 4.
	The Weka Experimenter software automatizes the process of running a set of machine learning methods, with various configurations on a series of training sets.
	
	It is used on the training sets in order to estimate the optimal value of the noise to use later with the *GaussianProcesses* method in the current situation.

• In the *Datasets* frame, click the button *Add* to add each of the files `train*_E_IIAB_R(2-4)_FC.arff` located in the CV folder. • In the *Algorithms* folder, click the button *Add new* and accept the default method (*ZeroR*) • Click the button *Add new* again. • In the pop up window, click the button *Choose*. • From the Weka hierarchy of machine learning methods, chose *GaussianProcesses* (Figure 13). • Set the *filter type* to *No normalization/standardization*. • Set the *noise* value to 1. • Click the button *OK*. • Repeat the process, add other Gaussian Processes, differing only by the *noise* value. Set the noise to integer values from 1 to 10. • Click the button *Save…* and create an experiment file called `ExtCVtrain.exp`. • Into the tab *Run* click the button *Start*.	The methods needs to decide the shape of a covariance function, that is identified to a kernel. Using ISIDA molecular descriptors, a simple but efficient choice is a linear kernel (the default value for the method) and no transformation of the descriptors' values. The configuration of the experiment is saved in a file named `ExtCVtrain.exp`. So, this setup can be reused at any time.

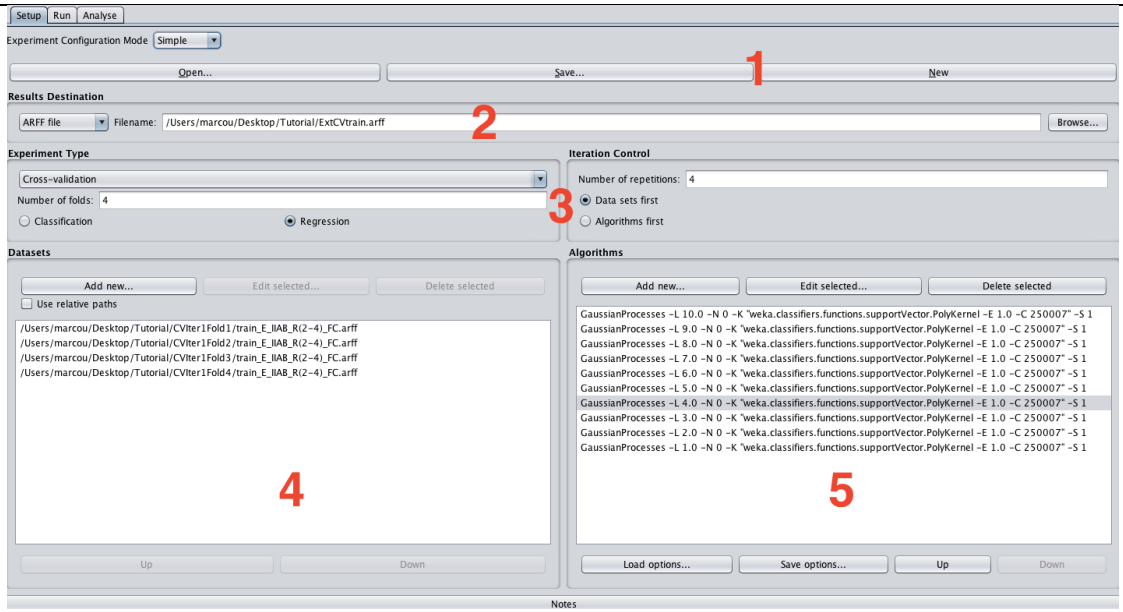

Figure 11: *Interface of Weka Experimenter. The top frame (1) is used to prepare the experiment: either prepare a new one, load or save an experiment. The frame below (2) is dedicated to saving the experiment report. The next frame (3) is the design of the experimental protocol: how model performances are evaluated and how many times the experiment is repeated. The bottom left frame (4) stores the datasets that are being processed and the bottom right frame (5) lists the machine learning methods to be experimented.*

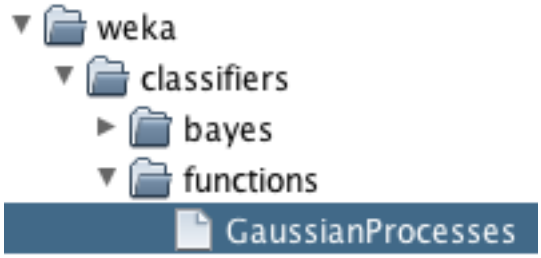

Figure 12: *Location of the Gaussian Processes into the hierarchy of Weka machine learning methods.*

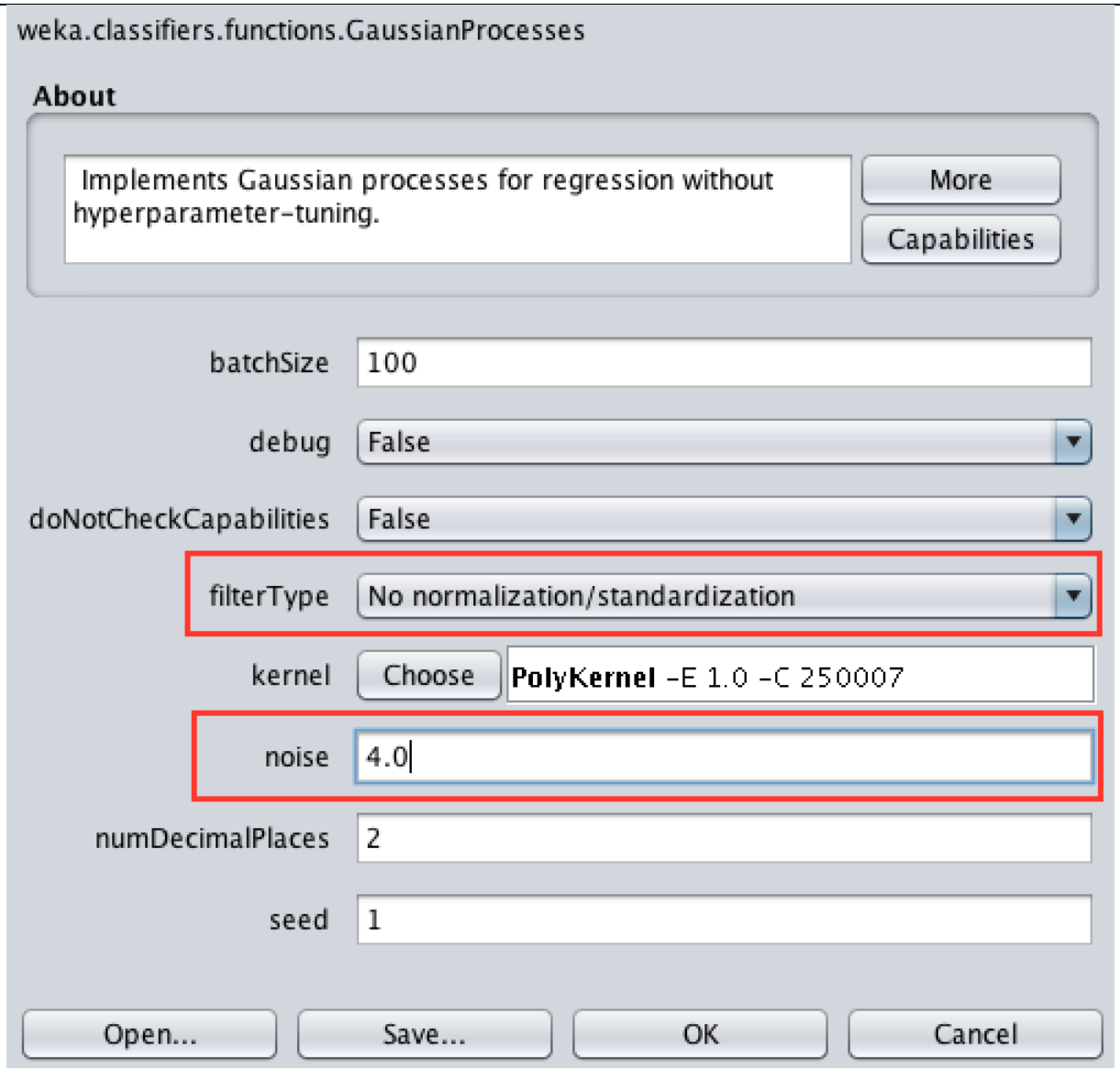

Figure 13: *Weka interface to configure a Gaussian Process method.*

- Click the *Analyse* tab to display the analysis interface of the experiment (Figure 14).
- Click the button *Experiment*.
- Set the *Comparison field* to *Relative_absolute_error*.
- Click the button *Perform test*.

The analysis interface displays the 640 results into a table. By default, lines are datasets and columns are methods.

The default method, *ZeroR*, should be the first one. This method consists in using as a regression method, the average of the property. In other words, using this model, all compounds are predicted the same value which is the average pKi value on 5-HT_{2B} as observed on the training set.

This analysis tool does pair comparison between the first method (here, *ZeroR*)

	and other methods (the Gaussian Processes). The symbols (v/ /*) are used to annotate those results that have larger, equivalent or lower value, respectively, compared to a base line classfier. Currently the base line classifier is the model *ZeroR* that consist into assigning the average pKi value to any instance. The bottom line summarizes the number of datasets for which the evaluated method got higher, equivalent or lower values compared to the reference. In the present case (Figure 15), the relative mean absolute error is used to check the models. Two conclusions emerge. First, the lowest errors are usually obtained for noise values about 3 or 4. However, some results are not annotated with an asterisk (*), meaning that they are no convicingly better than the base line model (*ZeroR).* These discrepancies are partly due to the small size of the dataset, to the low number of iterations in the experiment setup and to the presence of outliers. In the following the value of 4 is assumed to be the optimal noise level.

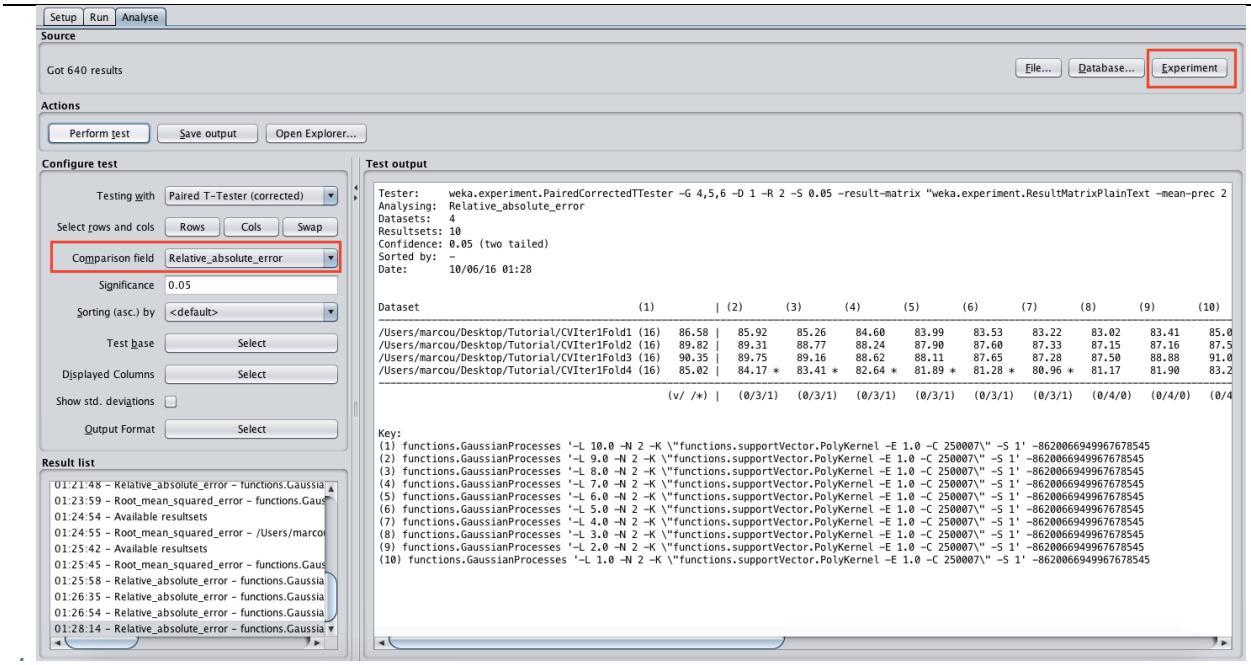

Figure 14: *Analysis interface of Weka Experimenter*

Dataset	(1) rules.Zer	(2) funct	(3) funct	(4) funct	(5) funct	(6) funct	(7) funct	(8) funct	(9) funct	(10) func	(11) func
CVIter1Fold1_affinity	(16) 100.00	86.58 *	85.92 *	85.26 *	84.60 *	83.99 *	83.53 *	83.22 *	83.02 *	83.41 *	85.01
CVIter1Fold2_affinity	(16) 100.00	89.82 *	89.31 *	88.77	88.24	87.90	87.60	87.33	87.15	87.16	87.55
CVIter1Fold3_affinity	(16) 100.00	90.35	89.75	89.16	88.62	88.11	87.65	87.28	87.50	88.88	91.09
CVIter1Fold4_affinity	(16) 100.00	85.02 *	84.17 *	83.41 *	82.64 *	81.89 *	81.28 *	80.96 *	81.17 *	81.90 *	83.23 *
	(v/ /*)	(0/1/3)	(0/1/3)	(0/2/2)	(0/2/2)	(0/2/2)	(0/2/2)	(0/2/2)	(0/2/2)	(0/2/2)	(0/3/1)

Key:
(1) rules.ZeroR '' 48055541465867954
(2) functions.GaussianProcesses '-L 10.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(3) functions.GaussianProcesses '-L 9.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(4) functions.GaussianProcesses '-L 8.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(5) functions.GaussianProcesses '-L 7.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(6) functions.GaussianProcesses '-L 6.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(7) functions.GaussianProcesses '-L 5.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(8) functions.GaussianProcesses '-L 4.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(9) functions.GaussianProcesses '-L 3.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(10) functions.GaussianProcesses '-L 2.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(11) functions.GaussianProcesses '-L 1.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545

Figure 15: *Example of results of Gaussian Processes with noise between 1 and 10, compared to a Zero rule model. The quantities tested are relative mean absolute errors.*

- In the *Weka Experimenter* software, click on the *Setup* tab.
- As shown on the area 1 in Figure 11, set the *Experiment Configuration Mode* to *Advanced*. The interface changes and should look like in Figure 16.
- If the Dataset area (area 4) is empty, use the button `Add new...`, to add the training set files. They are the files named `trainIter*Fold*.arff` in the `CV` folder.
- In the area 2, click on the button *New*.
- In the area 3, click on the default method *InstancesResultListener* to open the configuration interface, and change the name of the *outputFile* to `ExtCVtest.arff` inside the working directory.

In this setup, the models are prepared on the training set files. For each training set, the performances of the model is evaluated on the correspondong test set.

To automatize the evaluation of the models on their corresponding external test set, Weka expects that all test sets are located in the same directory (provided as the value of the *testsetDir* parameter). Second, the name of the test set file is computed as a concatenation of the directory, a prefix, a string derived from the name of the

- Set the value of *Runs* values from 1 to 1.
- Click the button *Choose* of the *Result Generator* frame and select *ExplicitTestsetResutProducer*.
- Click the word *ExplicitTestsetResutProducer* to open the configuration interface (Figure 17).
- Set the value of *RelationFind* to
  `(.*train|\.sdf.*)`
- Set the value of *testsetPrefix* to
  `test`
- Set the value of testsetSuffix to
  `_E_IIAB_R(2-4)_FC.arff`
- Set the value of *testsetDir* to
  `CV`
- Click the button *Choose* of the *splitEvaluator* frame and choose *RegressionSplitEvaluator*.
- Click the *RegressionSplitEvaluator* to open a new configuration interface (Figure 18).
- Click the button *Choose* of the *classifier* frame.
- Select the *GaussianProcesses* item (Figure 12).
- Click on the word *GaussianProcesses* to open the related configuration window.
- Configure the method as in Figure 13.
- Validate all the configuration interfaces by clicking on the buttons *OK*.
- Click the button *Save* and save your setup as `ExtCVtest.exp`.

relation (the keyword `@RELATION` in an ARFF file) and a suffix. The prefix and suffix are provided by the *testsetPrefix* and the *testsetSuffix* parameters.

The ISIDA/Fragmentor generates a relation name based on the name of the processed SDF file. To identify the test set file corresponding to a train set, the `Iter*Fold*` pattern is extracted. This is done using the regular expression provided as the *RelationFind* parameter. It will match all characters up to the word train and the final .sdf characters.

In this configuration, the *Runs* are individual evaluation procedures on all datasets. It can be a cross-validation procedure, so it can be repeated after shuffiling the datasets. In the present case, repeating the operations using the same train and test sets is useless : the results would be exactly the same for each run. This is why, it is set to 1.

Finally, it is a good idea to save the experimental setup.

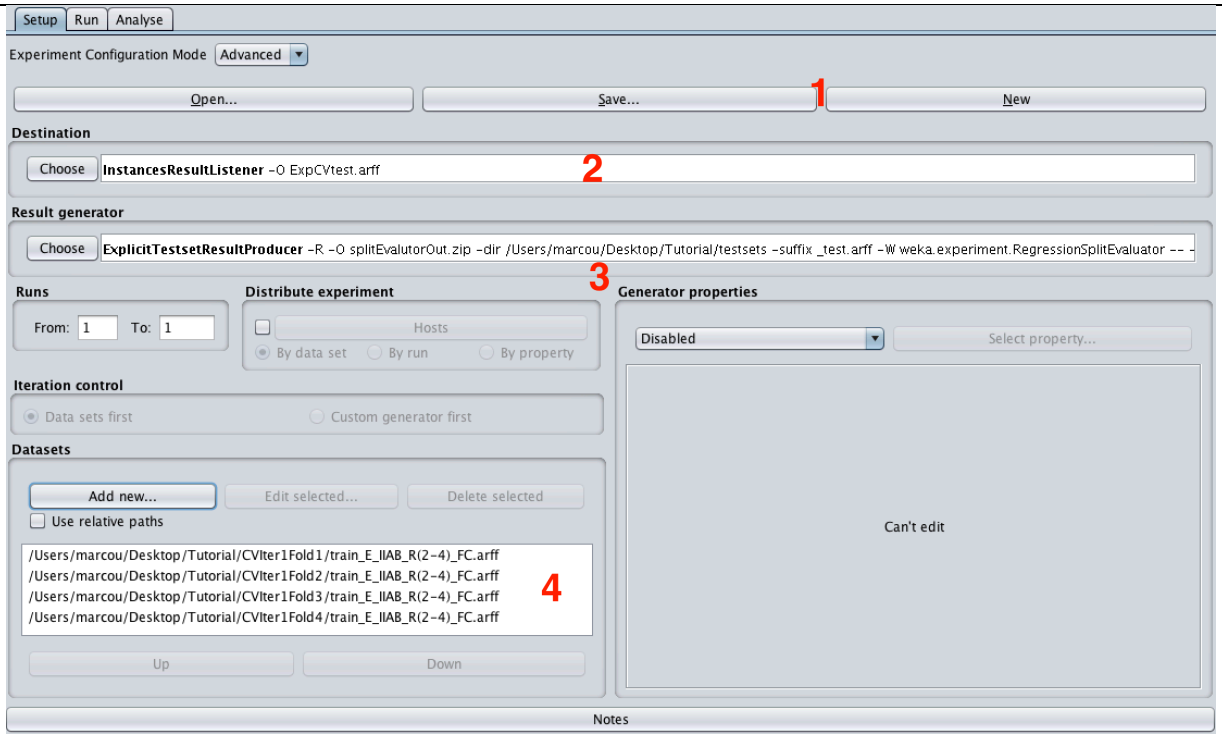

Figure 16: *Weka Experimenter Advanced mode configuration interface. The top frame (1) is used to prepare the experiment: either prepare a new one, load or save an experiment. The frame below (2) is dedicated to saving the experiment report. The next frame (3) is the design of the experimental protocol: it controls the runs of experiment, the type of model build on the training sets and how they are evaluated. The bottom left frame (4) stores the datasets that are being processed.*

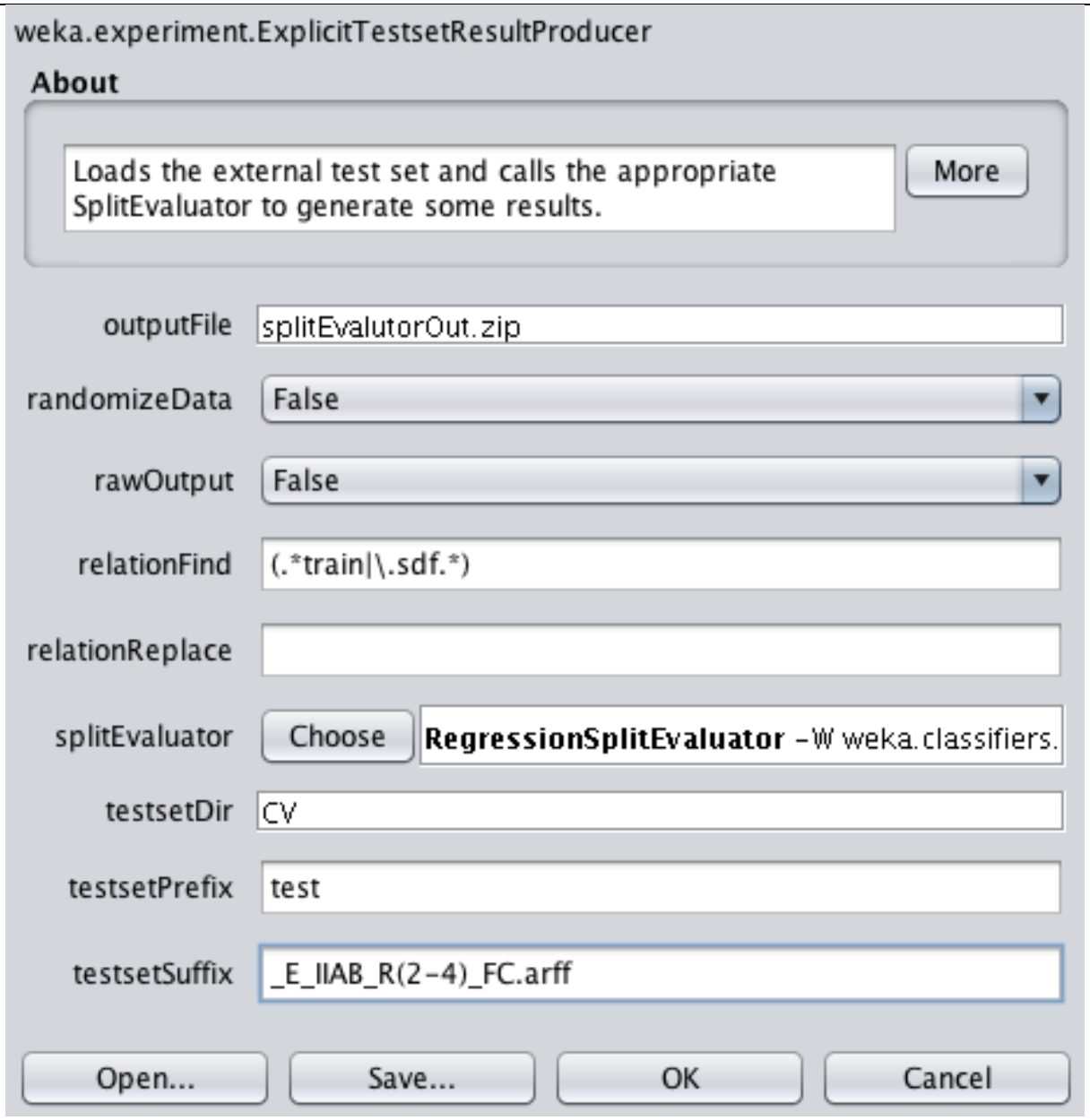

Figure 17: *Configuration interface of the ExplicitTestsetResultProducer.*

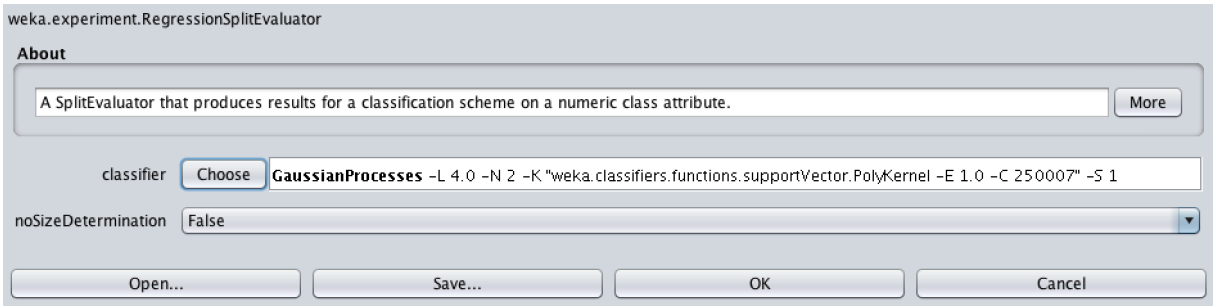

Figure 18: *Configuration interface of the RegressionSplitEvaluator.*

```
Datasets:   4
Resultsets: 1
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       10/06/16 15:22


Dataset                   (1) functions.
-----------------------------------------
CVIter1Fold1_affinity      (1)   80.30 |
CVIter1Fold2_affinity      (1)   74.91 |
CVIter1Fold3_affinity      (1)   73.11 |
CVIter1Fold4_affinity      (1)   84.02 |
-----------------------------------------
(v/ /*)                                |


Key:
(1) functions.GaussianProcesses '-L 4.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
```

Figure 19: *Example of Gaussian Processes performances on the external test set with a noise to a value of 4.*

- Click the *Run* tab.
- Click the button *Start*.
- Click the *Analyse* tab to display the analysis interface of the experiment (Figure 19).
- Click the button *Experiment*.
- Set the *Comparison field* to *Relative_absolute_error*.
- Click the button *Perform test*.

Typical results provided in Figure 19. Generally the results are improved compared to the cross-validation experiments (Figure 15). This is an expected result since now the entire train set is used for model fitting, in view of external cross-validation on the test set. By contrast, in the previous experiment, the training set itself underwent internal cross-validation, never being used as a whole for model fitting

Second, it is interesting to note that the datasets that performed best in internal cross-validation are those that generalized less in external cross-validation. This paradoxal result, the discrepancies during the optimization of the noise level of the Gaussian Processes, are clues that the dataset might contain some outliers.

Exercise 5:

The last part of the tutorial will focus on the identification of an outlier in the dataset. If some points are outliers, then they shall be difficult to fit by a non-overfitted model. The external cross-validation procedure basically identified a procedure to build a non-overfitted model on the dataset.

• Start the *Weka Explorer* software (Figure 20). • Click the button *Open file…* and load the file `IUPHAR_5HT2B_E_IIAB_R(2-4)_FC.arff`.	The full dataset is now loaded into the *Weka Experimenter* and will be studied in detail.

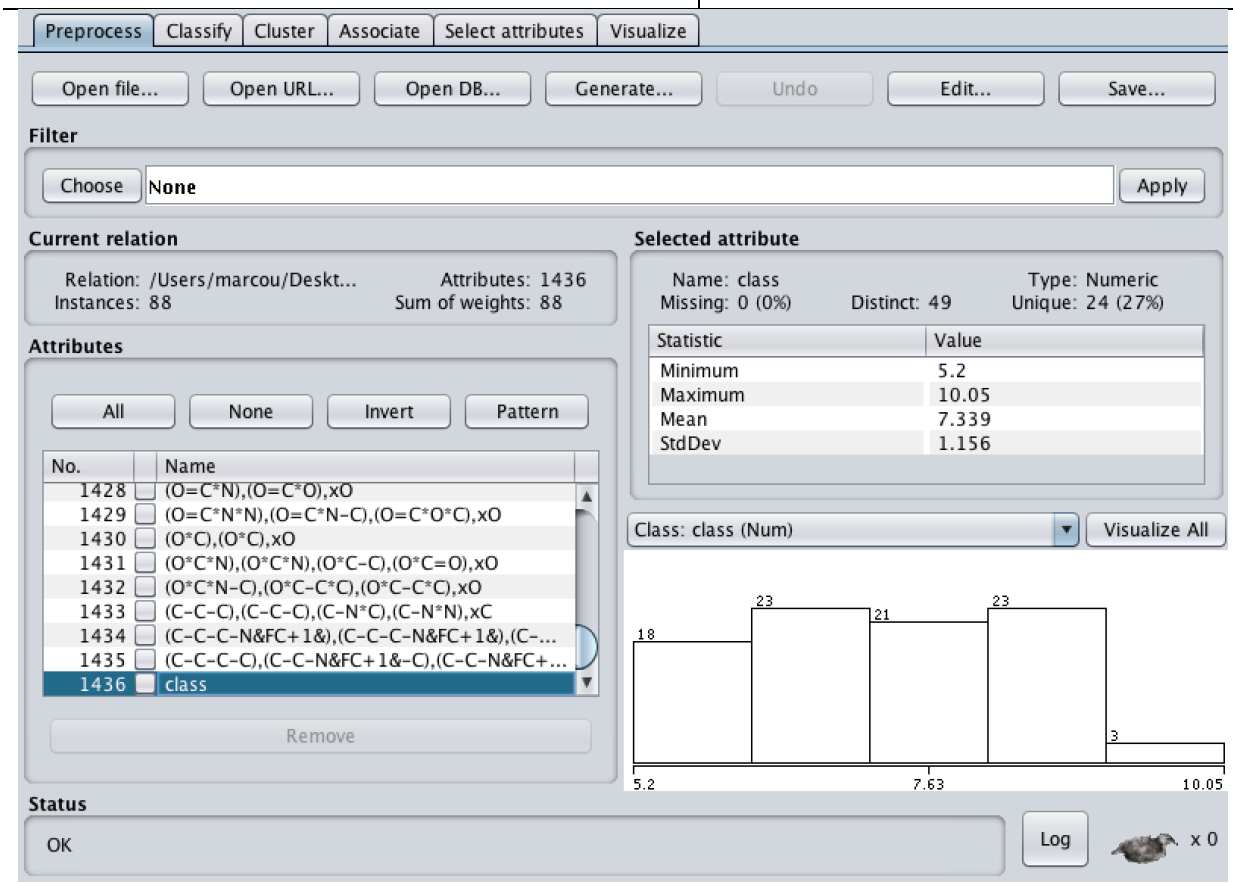

Figure 20: *The Weka Explorer software preprocessing mode.*

• Click the *Classify* tab (Figure 21). • Click the button *Choose* and select the *GaussianProcesses* item. • Click the *GaussianProcesses* word and set the configuration interface as in Figure 13. • Click the *Cross-validation* radio button and set the *Folds* value to 4. • Click the button *Start*.	During this procedure, the *GaussianProcesses* is used with the optimal setup identified with the *Weka Experimenter*. More detailed statistics are obtained (Figure 22). The performances are

	consistent with those obtained using the *Weka Experimenter* software.

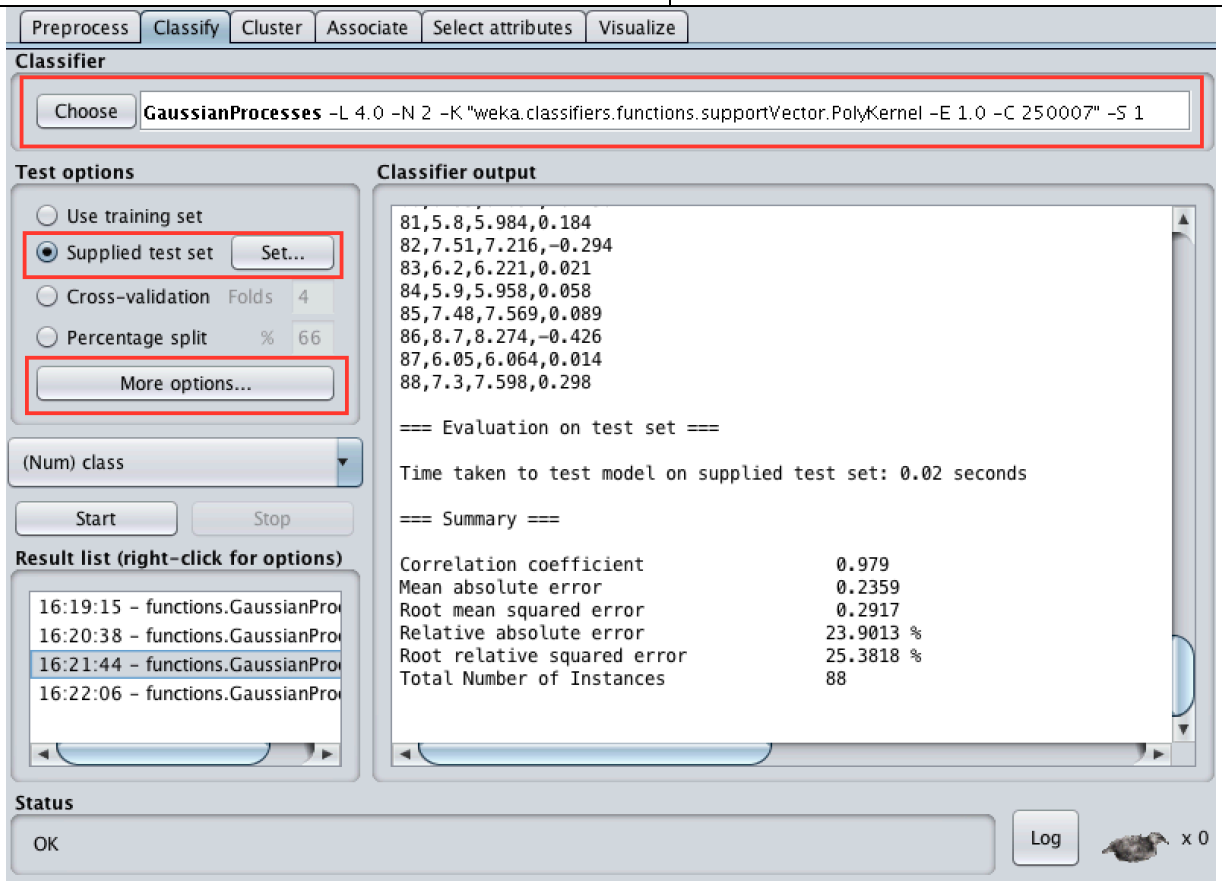

Figure 21: *Interface of the classification mode of Weka Explorer.*

```
Instances:    88
Attributes:   1436
              [list of attributes omitted]
Test mode:    4-fold cross-validation

=== Classifier model (full training set) ===

Gaussian Processes

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

All values shown based on: No normalization/standardization

Average Target Value : 7.3390909090909044
Inverted Covariance Matrix:
    Lowest Value = -0.01993988339279234
    Highest Value = 0.04610399059836405
Inverted Covariance Matrix * Target-value Vector:
    Lowest Value = -0.06391563323426866
    Highest Value = 0.04158533375446235



Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.5175
Mean absolute error                      0.7924
Root mean squared error                  0.9886
Relative absolute error                 79.8931 %
Root relative squared error             85.4517 %
Total Number of Instances               88
```

Figure 22: *Cross-validation results of Gaussian Processes in the Weka Explorer software.*

• Click the *Supplied test set* radio button then, the *Set* button. • Choose the file `IUPHAR_5HT2B_E_IIAB_R(2-4)_FC.arff` as test set file. • Click the button *More options…*(Figure 23). • In the configuration interface that opens, click the button *Choose* then select the *CSV* item. • Click the *Start* button.	During this procedure, the *GaussianProcesses* is used with the optimal setup identified with the *Weka Experimenter*. Some statistics are reported (Figure 24). The performances are consistent with those obtained using the *Weka*

• Right click on the line of the *Result list* corresponding to the fit results and select the option *Save result buffer*. • Name the output file `GP_Fit_all.out`.	*Experimenter* software. The estimated affinity value of each compound, along with the experimental value are provided into the log of the calculations. This output will be analyzed with another software, and therefore it must be saved into the file `GP_Fit_all.out`.

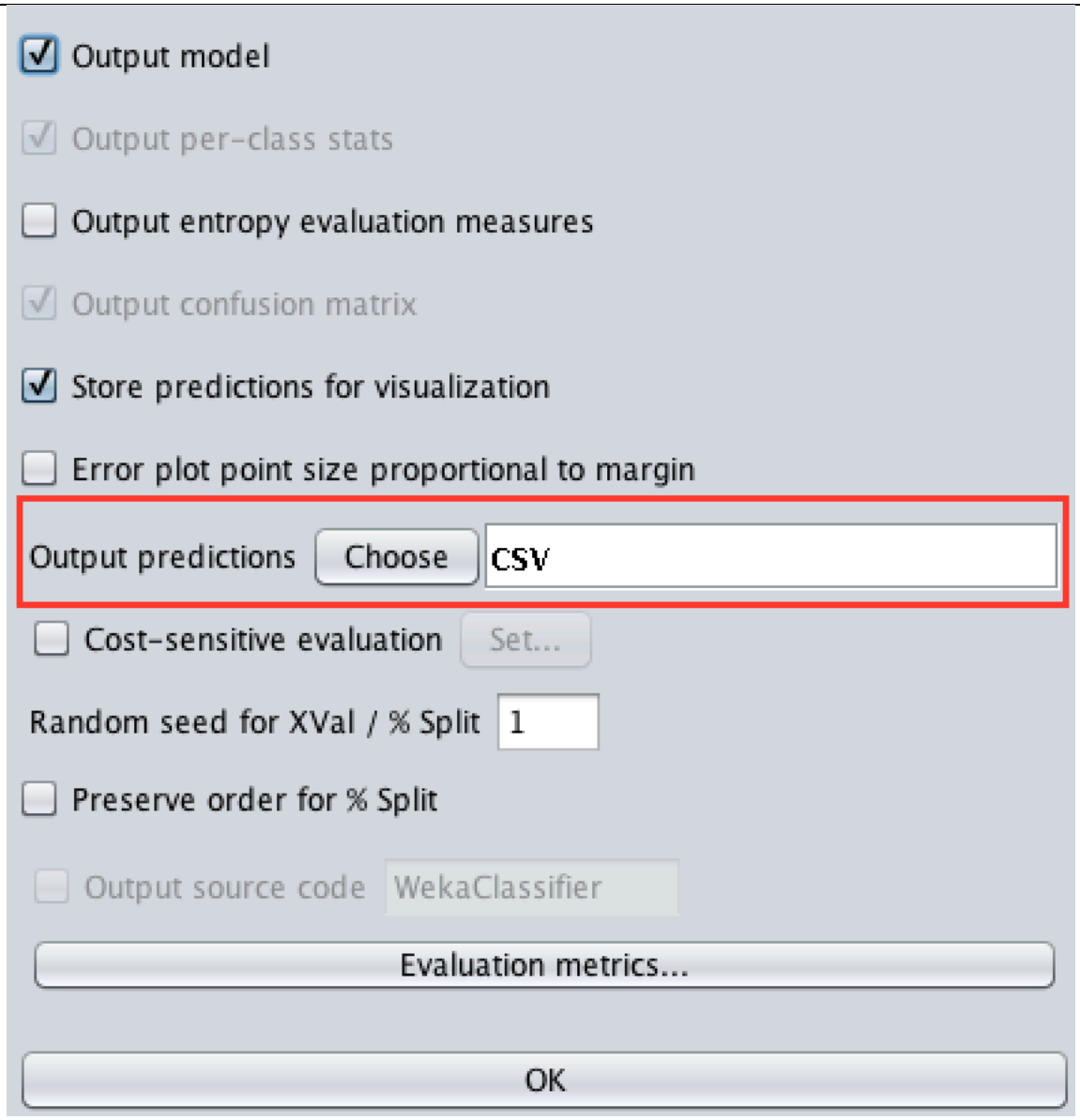

Figure 23: *The additional options of the classify tool of Weka Explorer.*

```
75,8.05,8.003,-0.047
76,5.9,6.059,0.159
77,6.8,6.677,-0.123
78,8.9,8.504,-0.396
79,5.8,6.114,0.314
80,8.95,8.692,-0.258
81,5.8,5.984,0.184
82,7.51,7.216,-0.294
83,6.2,6.221,0.021
84,5.9,5.958,0.058
85,7.48,7.569,0.089
86,8.7,8.274,-0.426
87,6.05,6.064,0.014
88,7.3,7.598,0.298

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correlation coefficient                  0.979
Mean absolute error                      0.2359
Root mean squared error                  0.2917
Relative absolute error                 23.9013 %
Root relative squared error             25.3818 %
Total Number of Instances               88
```

Figure 24: *Weka Gaussian Processes model fit statistics.*

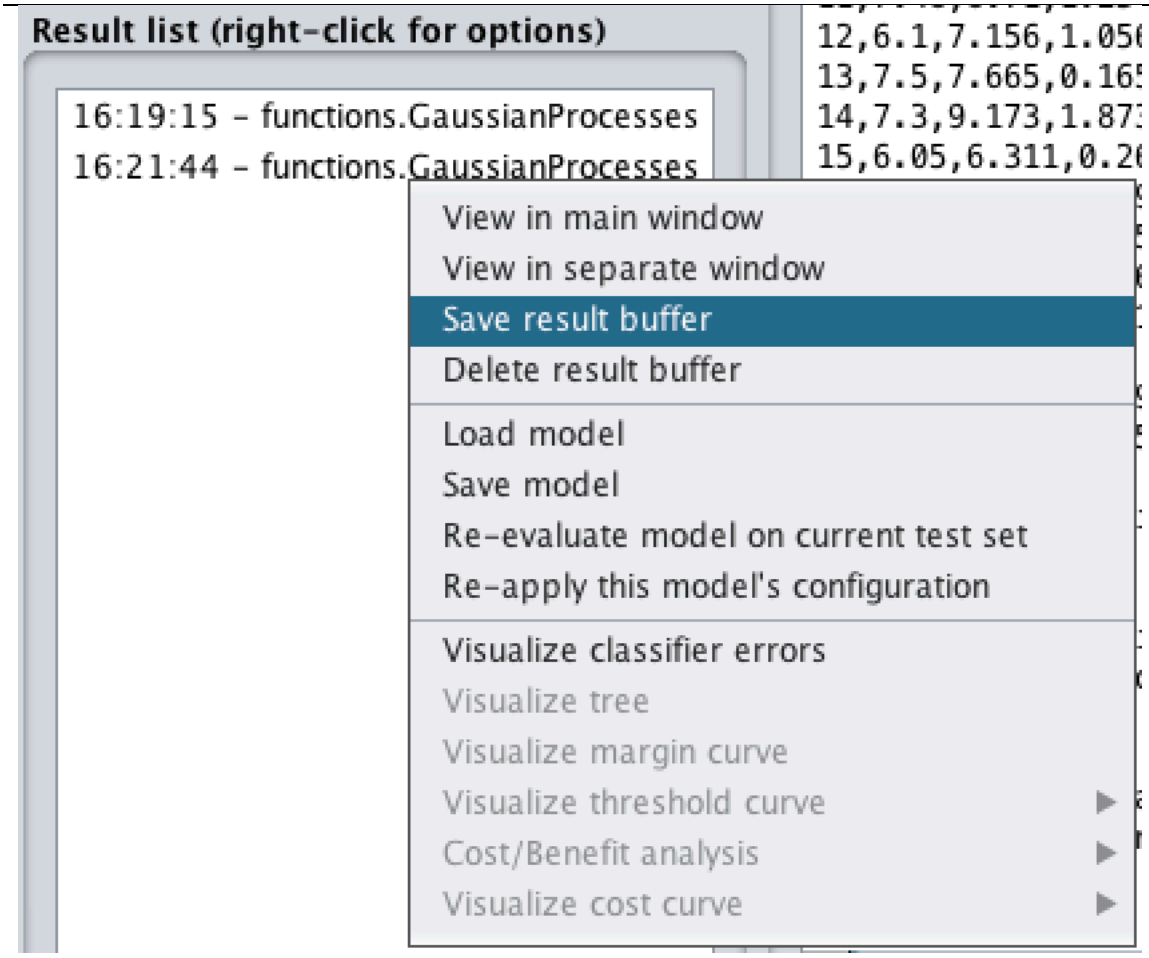

Figure 25: *To save the log of the calculation, right-click on the line of the Result list corresponding to the experiment to save, then select the Save result buffer option.*

• Open the file `GP_Fit_all.out` using the software *ModelAnalyzerR* (Figure 26). Use the file `IUPHAR_5HT2B.sdf` as the source of information on chemical structures. • Then click the *OK* button. • *Optional*. Search for the worst predicted compound and identify this compound in you *InstantJChem* database. • Optional. The outlier is Melatonin. You can search for its activity on 5HT2B on the IUPHAR database[1]. • Find the article referring to this molecule and search for a reason to keep or exclude the compound.	The ISIDA/ModelAnalyzerR software computes performance statistics of the models, provide interactive experimental versus estimated values and REC plots. Each point on the plots is linked the chemical structure information in the provided SDF file. It is easy to notice an outlier. The Grubbs test outputs a value of 3.4 located in between a risk of 5% and 1%. This is therefore a data point to consider. The point corresponds to the entry 19 of the database, melatonin (Figure 27). This value is determined in the reference [18], provided as the file

[1] http://www.guidetopharmacology.org/GRAC/ObjectDisplayForward?objectId=7, the 10/05/2017

	12750432.pdf. In this work the affinity of 5.2 of melatonin was reported by displacement of radiolabeled melusergine. However, on page 957 the authors stated: “Melatonin (…) partially attenuated the action of 5-HT, although only about 50% of inhibition was acquired even at a concentration of 100 μM. It was not possible, for reasons of solubility, to evaluate higher concentrations of melatonin.” In other words, the ligand is weak and measured at the limits of its solubility, in these conditions, the reported value is better understood as a qualitative, highlighting the importance of agomelatine. This is a good reason to exclude the entry from the study.

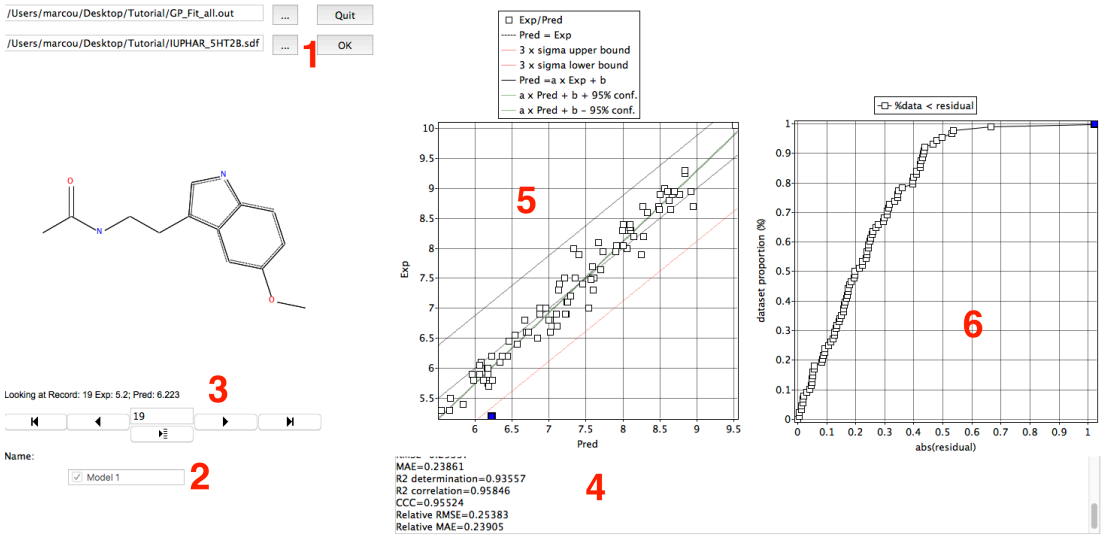

Figure 26: *Interface of the software ISIDA/ModelAnalyzerR. The names of input files are provided in area 1. If multiple models are available, they can be selected in area 2. Molecules are depicted in area 3. Statistical parameters are reported in the text box 4. The experimental versus estimated values are plotted in area 5. The REC is plotted in area 6.*

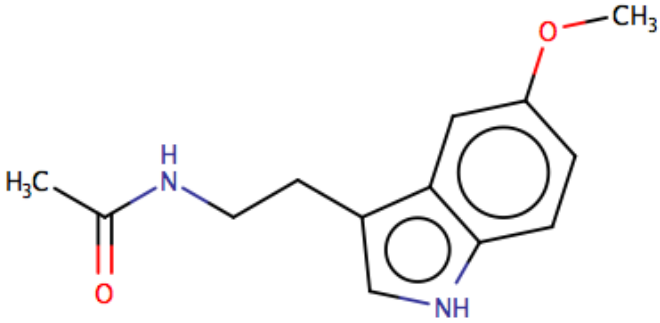

Figure 27: *Melatonin. The first outlier identified in the dataset. The database activity is 5.2.*

Optional. • In the *InstantJChem* database, get the *Original CdId* of the outlier. • Click the button *Query*. • In the Cell *Original CdId*, type the command "Not in list" followed by the *Original CdId* of the outlier. • Click the button *Run Query*. • Click the button Export to file (, Figure 8). Save the file in SDF format as `IUPHAR-5HT2B-33.sdf`, in the directory of the tutorial. • Click the button *Next* and be sure not to include the fields *CdId*, *Mol Weight* and *Formula*. • Click the button *Next* then *Finish*.	In this step a new dataset is created without the outlier chemical structure. The file `IUPHAR-5HT2B-33.sdf` is already prepared. It is possible to use this file directly if this step is skipped.
• In the *xFragmentor* software, click the	During this step the ISIDA SMF

button *Read XML*. Chose the file `IUPHAR_5HT2B_E_IIAB_R(2-4)_FC.xml`. • Remove all SDF files from the left hand side window with the *Remove SDF* button. • Click the *Add SDF* button and add the file `IUPHAR-5HT2B-33.sdf`. • Click the button *Run!*.	molecular descriptors are computed for this new dataset.
• Start the *Weka Explorer* software (Figure 20). • Click the button *Open file…* and load the file `IUPHAR-5HT2B-33_IIAB_R(2-4)_FC.arff`. • Click the *Classify* tab. • Click the button *Choose* and select the *GaussianProcesses* item. • Click the *GaussianProcesses* word and set the configuration interface as in Figure 13. • Click the *Cross-validation* radio button and set the *Folds* value to 4. • Click the button *Start*. • Click the *Supplied test set* radio button then, the *Set* button. • Chose the file `IUPHAR_5HT2B-33_E_IIAB_R(2-4)_FC.arff` as test set file. • Click the button *More options…*(Figure 23). • In the configuration interface that opens, click the button *Choose* then select the *CSV* item. • Click the *Start* button. • Right click on the line of the *Result list* corresponding to the fit results and select the option *Save result buffer*. • Name the output file `GP_Fit_all-33.out`.	A new *GaussianProcesses* is build using the *Weka* software. The model building uses the same optimal parameterization as discovered previously with *Weka Experimenter*. During the cross-validation procedure, the performances of the model are enhanced. At the fitting stage the same observation is made.

```
=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.5157
Mean absolute error                      0.7881
Root mean squared error                  0.9771
Relative absolute error                 78.6143 %
Root relative squared error             83.9261 %
Total Number of Instances               87
```

Figure 28: *Cross-validation results of Gaussian Processes, without the melatonin data.*

```
=== Summary ===

Correlation coefficient                  0.9818
Mean absolute error                      0.2299
Root mean squared error                  0.2705
Relative absolute error                 23.6221 %
Root relative squared error             23.881  %
Total Number of Instances               87
```

Figure 29: *Gaussian Process fit statistics, without melatonin.*

• Open the file `GP_Fit_all-33.out` using the software *ModelAnalyzerR*. Use the file `IUPHAR_5HT2B-33.sdf` as the source of information on chemical structures. • Then click the *OK* button.	Now, there are no compounds that can be identified as an outlier, or even be suspected to be an outlier.

Conclusion

During this tutorial, a very cautious workflow is illustrated, using external cross-validation to assess the predictive power of the QSAR model generated and internal cross-validation to estimate the optimal parameters of the machine learning method used. These optimal parameters are a guarantee that the models will be neither over-fitted nor under-fitted.

During the process, it was suspected that some point of the initial data could be pathological. The machine learning method with optimal parameters is used to fit the training set. The outlier data should be the data point most difficult to fit. Removing this point, the performances of the model are improved.

An important point, not illustrated by this tutorial, is that outliers may also depend on the molecular descriptors and of the machine learning method used. Therefore, an obvious solution to make outlier detection more robust is to repeat the procedure with different molecular descriptors and different machine learning algorithms.

Using this strategy, it is possible to prioritize several suspicious points. Yet, if the content of the dataset has changed because of the removing of one or more outliers, the whole procedure of internal cross-validation optimization of the machine learning parameters and of the external cross-validation estimate of generalization should be repeated.

References

1. Southan, C., et al., *The IUPHAR/BPS Guide to PHARMACOLOGY in 2016: towards curated quantitative interactions between 1300 protein targets and 6000 ligands.* Nucleic acids research, 2015: p. gkv1037.
2. Barnes, N.M. and J.F. Neumaier, *Neuronal 5-HT receptors and SERT.* Tocris bioscience scientific review series, 2011. **34**: p. 1-15.
3. Roth, B.L., *Drugs and valvular heart disease.* N Engl J Med, 2007. **356**(1): p. 6-9.
4. Rothman, R.B., et al., *Evidence for possible involvement of 5-HT2B receptors in the cardiac valvulopathy associated with fenfluramine and other serotonergic medications.* Circulation, 2000. **102**(23): p. 2836-2841.
5. Connolly, H.M., et al., *Valvular heart disease associated with fenfluramine–phentermine.* New England Journal of Medicine, 1997. **337**(9): p. 581-588.
6. *FDA Announces Withdrawal Fenfluramine and Dexfenfluramine (Fen-Phen)*, U.S.F.A.D. Administration, Editor. 1997.
7. Frachon, I., et al., *Benfluorex and unexplained valvular heart disease: a case-control study.* PloS one, 2010. **5**(4): p. e10128.
8. Rodrigo Andrade, N.M.B., Gordon Baxter, Joel Bockaert, Theresa Branchek, Marlene L. Cohen, Aline Dumuis, Richard M. Eglen, Manfred Göthert, Mark Hamblin, Michel Hamon, Paul R. Hartig, René Hen, Katharine Herrick-Davis, Rebecca Hills, Daniel Hoyer, Patrick P. A. Humphrey, Klaus Peter Latté, Luc Maroteaux, Graeme R. Martin, Derek N. Middlemiss, Ewan Mylecharane, Stephen J. Peroutka, Pramod R. Saxena, Andrew Sleight, Carlos M. Villalon, Frank Yocca. *5-Hydroxytryptamine receptors: 5-HT2B receptor.* IUPHAR/BPS Guide to PHARMACOLOGY 2016 12/08/2015 08/06/2016]; Available from: http://www.guidetopharmacology.org/GRAC/ObjectDisplayForward?objectId=7.
9. *PubMed*. U.S. National Library of Medecine 08/06/2016]; Available from: http://www.ncbi.nlm.nih.gov/pubmed/?term=.
10. MacKay, D.J., *Introduction to Gaussian processes.* NATO ASI Series F Computer and Systems Sciences, 1998. **168**: p. 133-166.
11. Rasmussen, C.E., *Gaussian processes for machine learning.* 2006.
12. Neal, R.M., *Monte Carlo implementation of Gaussian process models for Bayesian regression and classification.* arXiv preprint physics/9701026, 1997.
13. Maimon, O. and L. Rokach, *Data mining and knowledge discovery handbook*. Vol. 2. 2005: Springer.
14. Aggarwal, C.C. *Outlier analysis*. in *Data Mining*. 2015. Springer.
15. Ruggiu, F., et al., *Quantitative Structure-Property Relationship Modeling: A Valuable Support in High-Throughput Screening Quality Control.* Analytical Chemistry, 2014. **86**(5): p. 2510-2520.
16. Grubbs, F.E., *Sample criteria for testing outlying observations.* The Annals of Mathematical Statistics, 1950: p. 27-58.
17. Pearson, E.S. and C.C. Sekar, *The efficiency of statistical tools and a criterion for the rejection of outlying observations.* Biometrika, 1936. **28**(3/4): p. 308-320.
18. Millan, M.J., et al., *The novel melatonin agonist agomelatine (S20098) is an antagonist at 5-hydroxytryptamine2C receptors, blockade of which enhances the activity of frontocortical dopaminergic and adrenergic pathways.* Journal of Pharmacology and Experimental Therapeutics, 2003. **306**(3): p. 954-964.