

# Tutorial:

# From chemical structures to outlier analysis

G. Marcou, D. Horvath, A. Varnek



# Download your materials

- **Download URL (tinyURL to Renater FileSender repository):**
  - ✓ <https://tinyurl.com/y99km3nd>
- **Content (<150Mo when unzipped)**
  - ✓ Softwares
    - [ExtCV.exe](#), [xFragmentor.exe](#), [xModelAnalyzerR.exe](#)
  - ✓ Datasets
    - **Datamining**
      - 5-HT2B (a database)
      - Iter0 (tutorial directory)
      - Iter1 (result after the tutorial)
    - **Pubmed**
      - Relevant articles

# Introduction

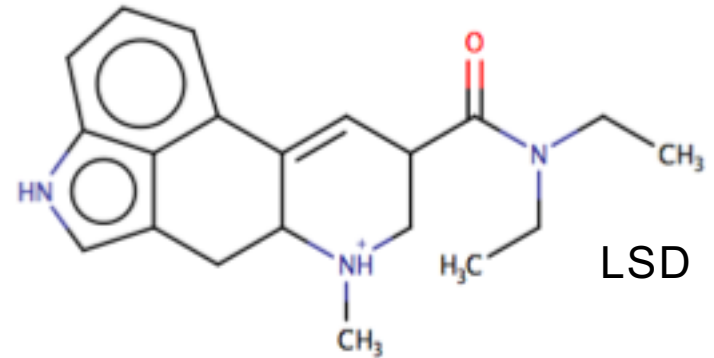
## ■ The IUPHAR 5-HT<sub>2B</sub> dataset

- ✓ 88 compounds
- ✓ Affinity values
  - pK<sub>i</sub>, pK<sub>d</sub>
  - Mostly human
- ✓ Type
  - Agonist
  - Antagonist
- ✓ Bibliography
  - PubMed Ids

# 5-HT2A

Expressed in forebrain  
cerebral cortex

Famous 5-HT2A agonist



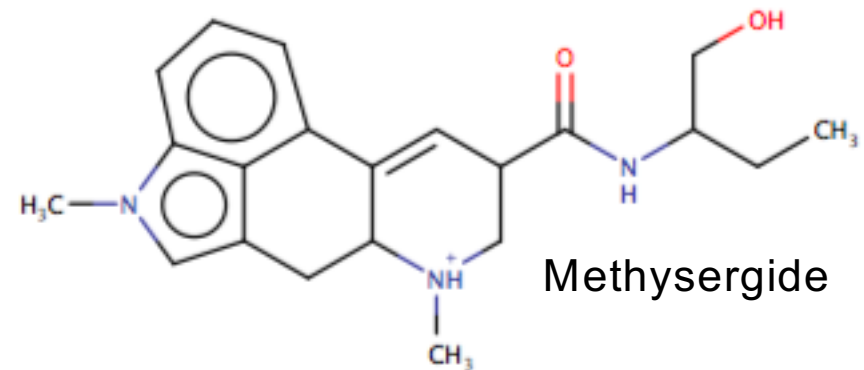
**IMPLICATED IN  
FLOWER POWER**

# 5-HT<sub>2C</sub>



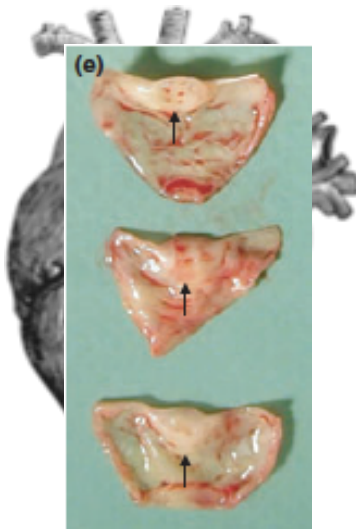
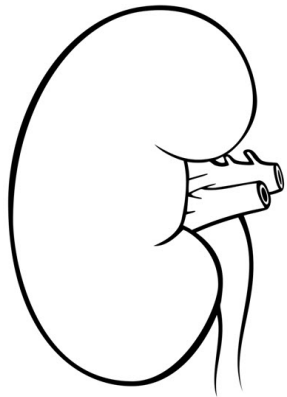
Expressed in CNS

Example 5-HT<sub>2C</sub>  
antagonist



Implicated into:  
obesity, seizure, psychotic  
disorder

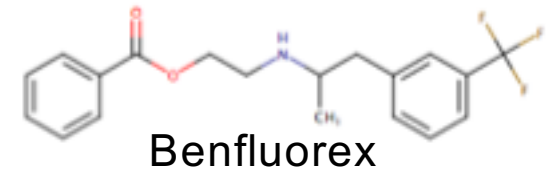
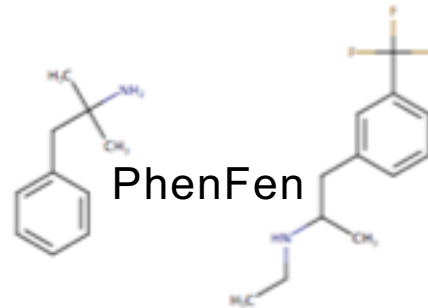
# 5-HT<sub>2B</sub>, the anti-target



Expressed into

- Liver
- Kidney
- Heart

Example ligands:



Drug induced  
valvular heart  
disease

# Exercise 1

- **Goal:**
  - ✓ Create a database of 5-HT2B ligand
- **Software:**
  - ✓ InstantJChem
- **File**
  - ✓ IUPHAR\_5HT2B.sdf
- **Output**
  - ✓ Directory 5-HT2B

# Create a local database

1. Choose Project
2. ...

Instant JChem



Categories:

 General

Projects:

- IJC Project (empty)
- IJC Project (with local database)
- IJC Project (local database with demo data)

Description:

New Instant JChem project with local database



# Configure local database

## Steps

1. Choose Project
2. IJC Project Name

## IJC Project Name

Project Name:

Project Location:

[Browse...](#)

Project Folder:

Close Already Opened Projects

Instant JChem



Help

< Back

Next >

Finish

Cancel

# Load an SDF into the database

## Steps

1. Select schema
2. **File and new table details**
3. Field details
4. Monitor import

## File and new table details

Database:  localdb

File to import:

File type:

Table details:

Summary:

### 10 fields found:

```
Structure [Text,Structure,List (Text)]
Cdid [Text,List (Text),Boolean,Decimal,Integer]
Mol Weight [Text,List (Text),Decimal]
Formula [Text,List (Text)]
target_species [Text,List (Text)]
ligand [Text,List (Text)]
type [Text,List (Text)]
affinity_units [Text,List (Text)]
affinity [Text,List (Text),Decimal]
pubmed_id [Text,List (Text)]
```

Records read:

Instant JChem



# SDF field management

## Steps

1. Select schema
2. File and new table details
3. **Field details**
4. Monitor import

## Field details

### Fields in file

- Structure
- Cdid
- Mol Weight
- Formula
- target\_species
- ligand
- type
- affinity\_units
- affinity
- pubmed\_id

Add >

< Merge >

< Map >

< Remove

Move up

Move down

### Fields in database

- 123 Cdid
- + Structure [Structure]
- 1,23 Mol Weight
- A Formula
- + A target\_species
- + A ligand
- + A type
- + A affinity\_units
- + 1,23 affinity
- + A pubmed\_id
- + 123 Original Cdid [Cdid]

New field type: 123 Integer Field

Display name: Original Cdid

Required: FALSE

Default value:

DB Column Name: Original\_Cdid

Instant JChem



# Grid view column management

The image shows a software interface for managing columns in a grid view. A context menu is open over a table header 'Cdid', with the following options:

- New Standard Field
- New Chemical Term
- New Calculated Field
- New Row... Data
- ✓ Fit table to screen
- Adjust columns width
- Open Column Management**
- Customize Widgets

The background shows a dialog box for field selection. It has two main sections: 'Available fields' and 'Selected fields'.

**Available fields:**

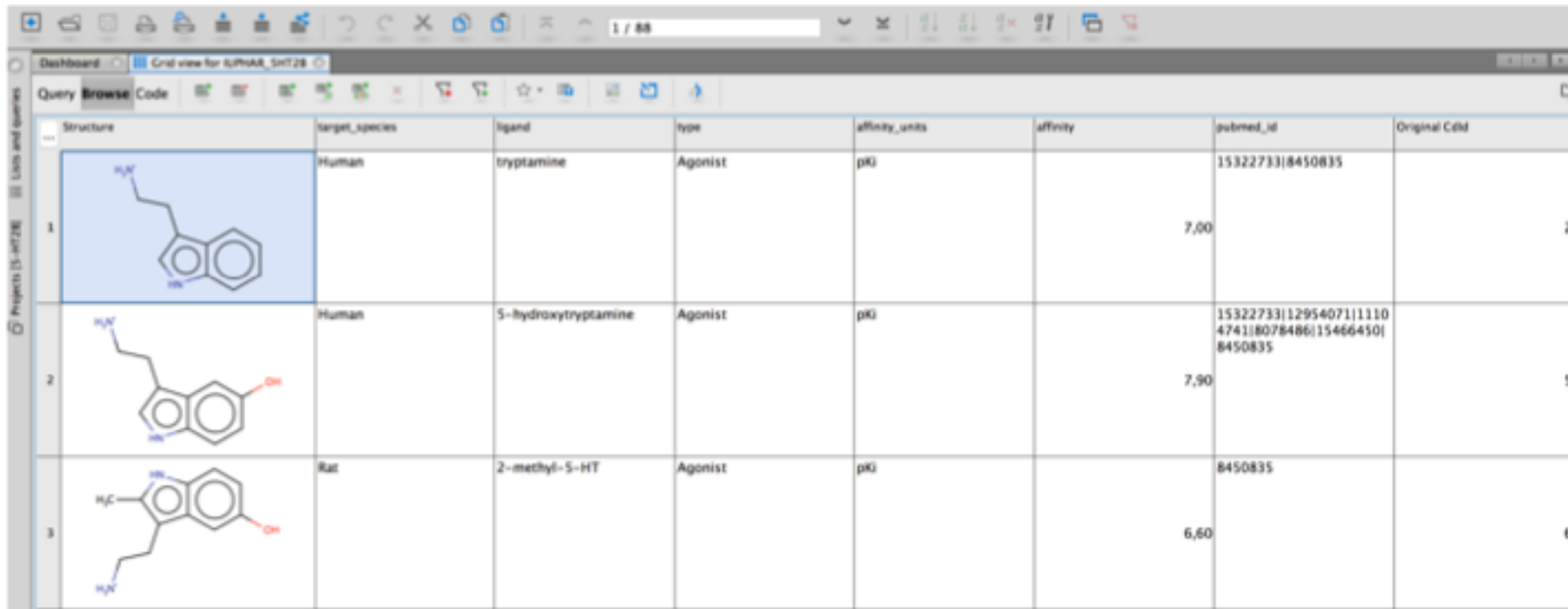
- 123 Cdid
- 1,23 Mol Weight
- A Formula

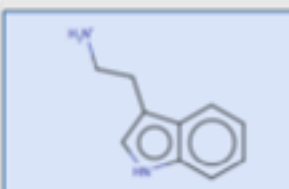
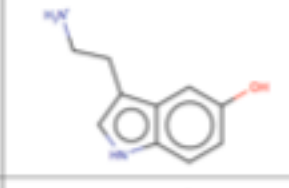
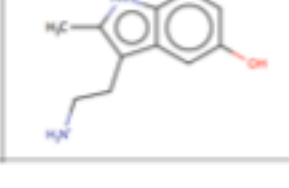
**Selected fields:**

- Structure
- A target\_species
- A ligand
- A type
- A affinity\_units
- 1,23 affinity
- A pubmed\_id
- 123 Original Cdid

Buttons in the dialog include: Add ->, Add All ->, <- Remove, <- Remove All (highlighted), Move Up, Move Down, Cancel, and OK.

# Final state of the interface



Structure	target_species	ligand	type	affinity_units	affinity	pubmed_id	Original CellId
	Human	tryptamine	Agonist	pK <sub>i</sub>		15322733 8450835	
	Human	5-hydroxytryptamine	Agonist	pK <sub>i</sub>		15322733 12954071 11104741 8078486 15466450 8450835	
	Rat	2-methyl-5-HT	Agonist	pK <sub>i</sub>		8450835	

- A database of 5-HT<sub>2B</sub> ligands
- Access to chemical structures, affinity, type, identifier, bibliographic references
- Possibility to edit the dataset: structures, instances and values

# Exercise 2

- **Goal:**

- ✓ Set up an External Cross-Validation procedure

- **Software:**

- ✓ ExtCrossValidate
- ✓ xFragmentor

- **File**

- ✓ IUPHAR\_5HT2B.sdf

- **Output**

- ✓ Directories CVIteriFoldj, files train.sdf and test.sdf
- ✓ ISIDA Substructural Molecular Fragments, files \*.hdr, \*.svm, \*.arff

# Software: ExtCrossValidate

Choose an SDF file:

1 Mining/Iter0/IUPHAR\_5HT2B.sdf ...

3 k= 1 N= 4 2 Cross Validation

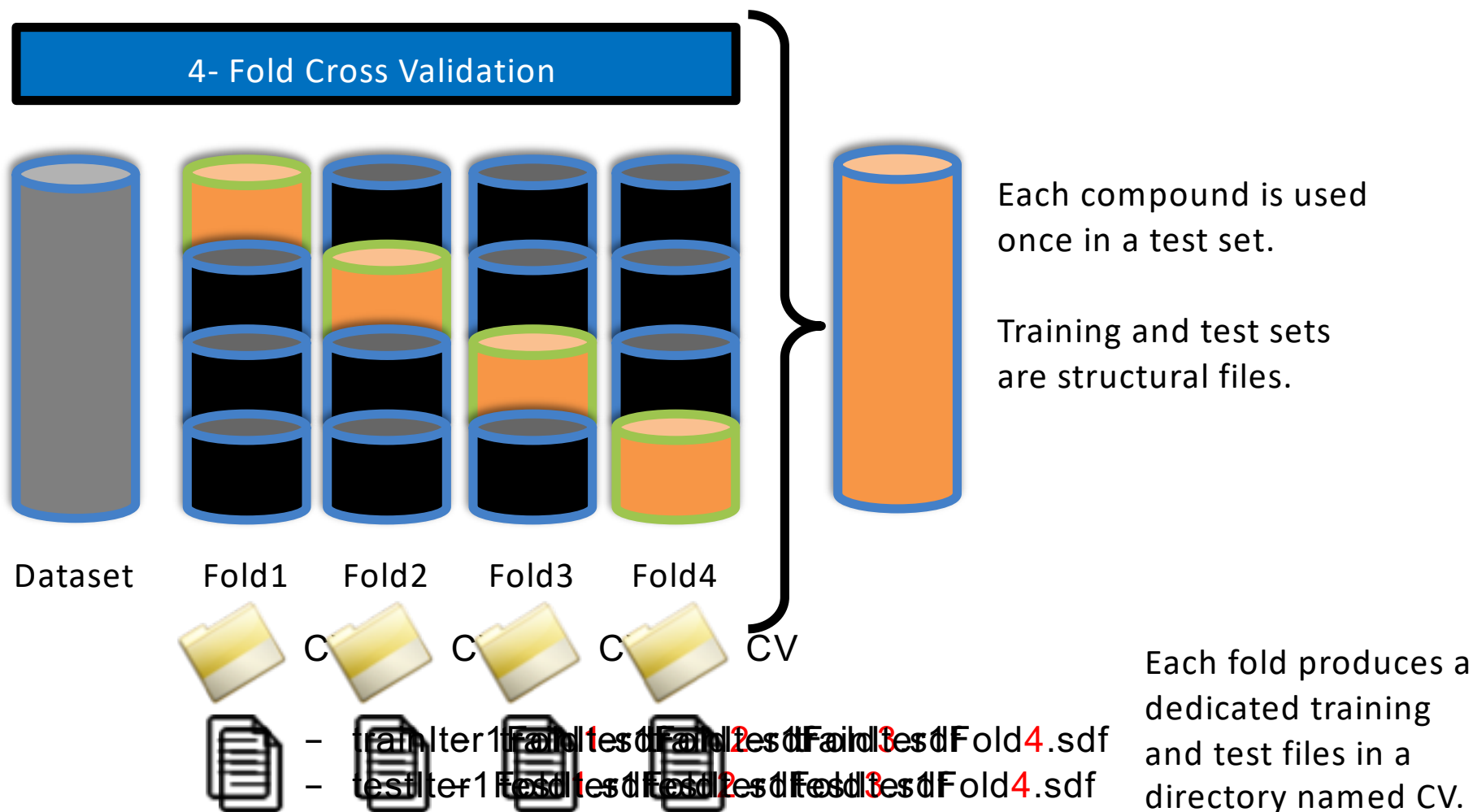
Create separated folders 4

Quit

Run!

- 1 Select the **IUPHAR\_5HT2B.sdf** file
- 2 Choose **Cross Validation**
- 3 Set 1 iteration to 1: **k=1** and 4 folds: **N=4**
- 4 **Uncheck** the tick-box. Store all train and test sets into a folder named CV

# External Cross-Validation





# Software: xFragmentor

List of SDF files to process:

ktop/Tutorial/CViter1Fold1/train.se  
ktop/Tutorial/CViter1Fold2/train.se  
ktop/Tutorial/CViter1Fold3/train.se  
ktop/Tutorial/CViter1Fold4/train.se  
ktop/Tutorial/IUPHAR\_5HT2B.sdf

Add SDF

Remove SDF

SDF field of interest:

affinity

Select a topology:

IIR (Atom centered, homogeneous path s

Add Fragment

Min length

Max length

2

4

Delete Fragment

2

Select a coloration scheme for atoms:

A Atom type

Select a coloration scheme for bonds

B Bond type

Read XML

Save XML

Atom Pairs

Do All Ways

Use Formal Charges

Use predefined fragments

Base name of header files...

Check

Use only those fragments

3

0 Do not use marked atom

Get atom contribution to fragment

E  
IIAB\_R(2-4)\_FC

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ISIDA/xFragmentor
Universit  de Strasbourg, 2016
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Quit

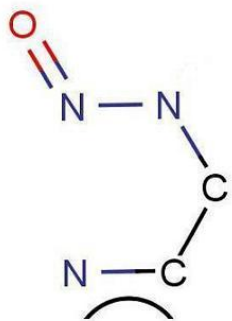
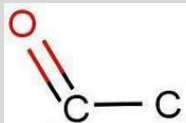
Reset

Run!

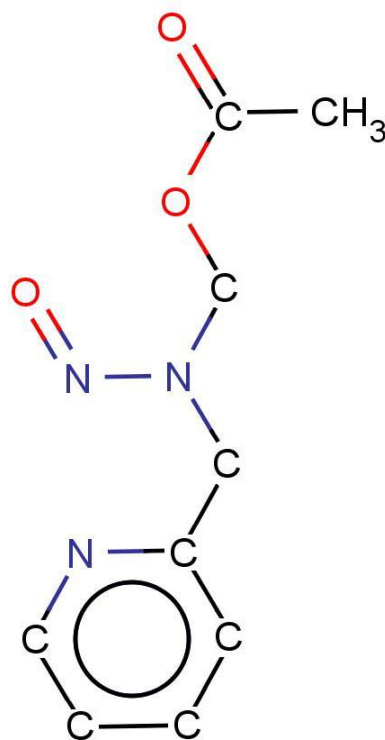
# ISIDA fragments

Sequences of atoms and bonds containing

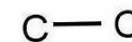
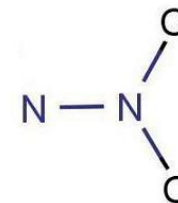
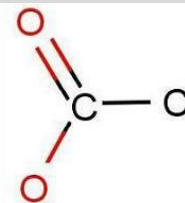
$N_{\min} > 2$  to  $N_{\max} < 15$  atoms



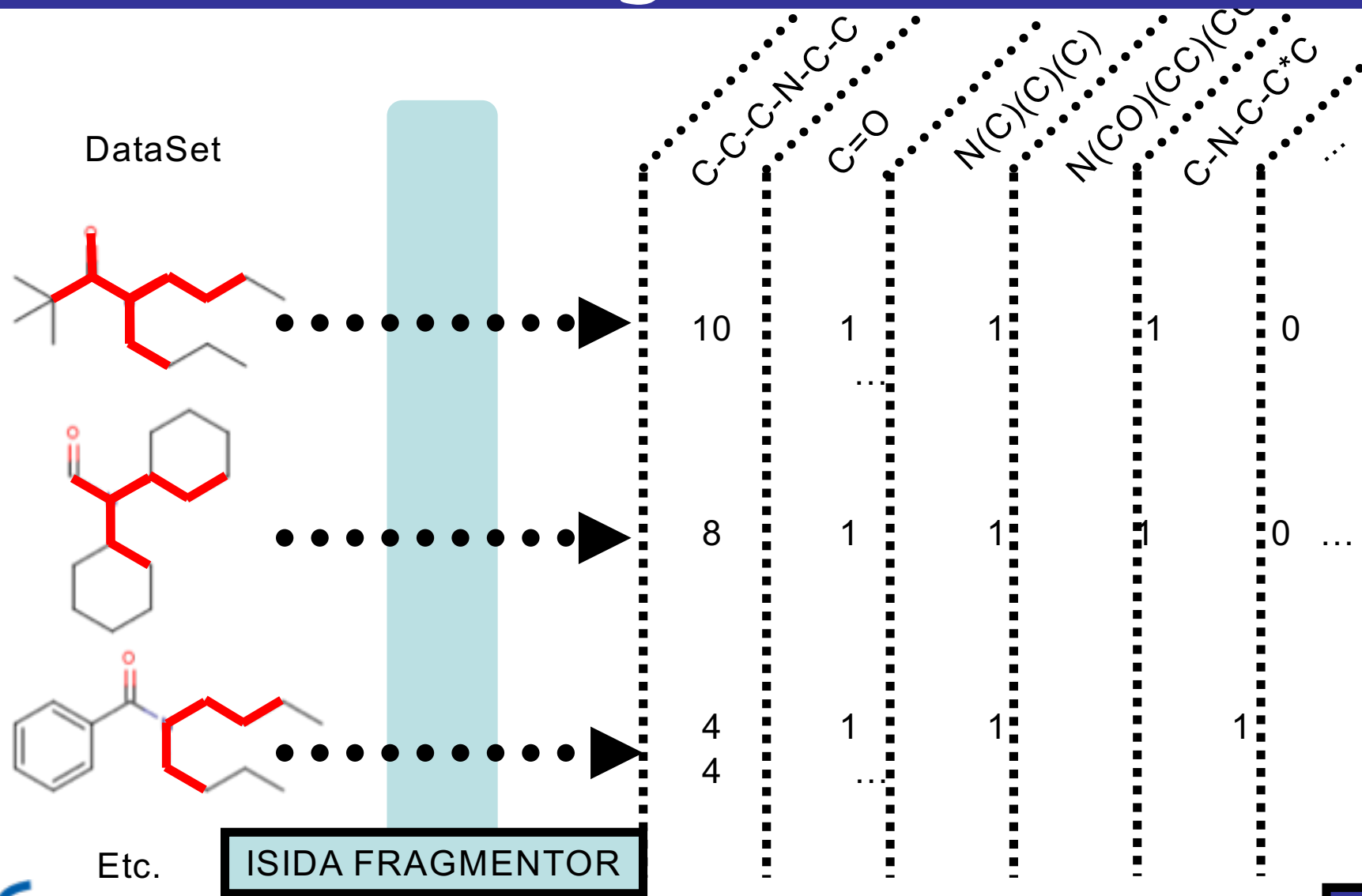
*atoms and bonds*



Augmented atoms  
(a central atom and his  
direct neighbours)



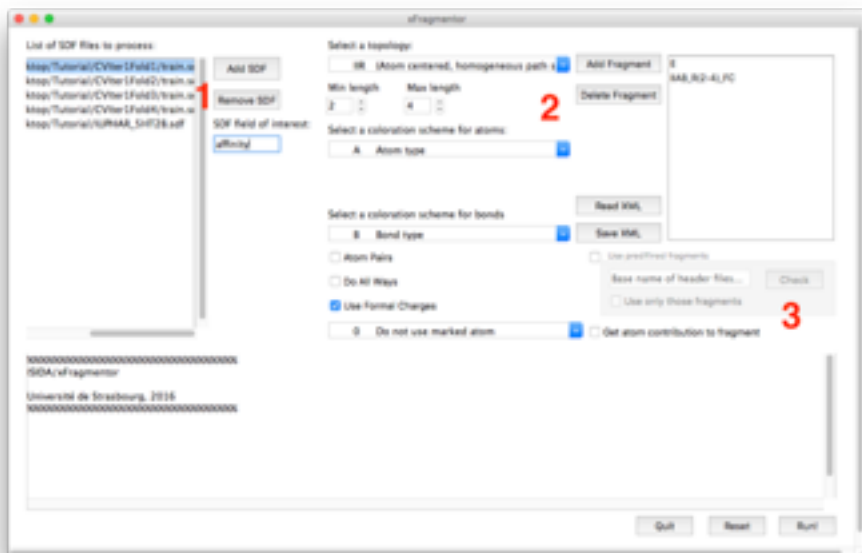
# ISIDA Substructural Molecular Fragments



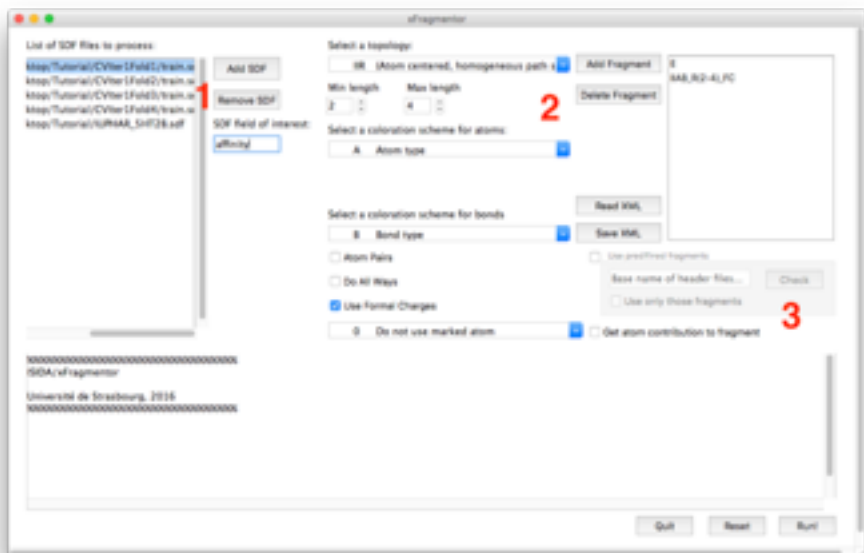
the Pattern matrix

# Compute training set descriptors

- **Add SDF files (area 1)**
  - ✓ IUPHAR\_5-HT2B.sdf
  - ✓ CV/trainIteriFoldj.sdf
- **Add Fragmentations (area 2)**
  - ✓ Atom count: E
  - ✓ Atom centered of length 2 to 4: IIAB(2-4)
  - ✓ Use formal charge: FC
- **Save configuration**
  - ✓ Click the *Save XML* button
  - ✓ Name the file train\_E\_IIAB\_R(2-4)\_FC.xml
- **Click the *Run* button**



# Compute test set descriptors



- **Remove all SDF files and add test SDF files (area 1)**
  - ✓ Remove IUPHAR\_5-HT2B.sdf
  - ✓ Remove CV/trainIteriFoldj.sdf
  - ✓ Add CV/testIteriFoldj.sdf
- **Check the tickbox Use Predefined Fragments (area 3)**
  - ✓ Leave Basis name to default value or empty
- **Use the Check button (optional)**
  - ✓ You should receive the message *All hear files were found.*
- **Save configuration**
  - ✓ Click the *Save XML* button
  - ✓ Name the file test\_E\_IIAB\_R(2-4)\_FC.xml
- **Click the *Run* button**

# Files created by xFragmentor

- XML:  
Configuration file.

```
<?xml version="1.0" encoding="u
<Fragmentation Max="2" Min="2" Atom
AtomFrg="False" Extended="NoFX" Mar
AtomPairs="False" DoAllWays="False"
```

- SVM:  
Molecular descriptors (attributes)/  
Property value (value) files in  
sparse format.  
Each molecule correspond to one  
line, in order of appearance in the  
SDF file.

- ARFF:  
Molecular descriptors (attributes)/  
Property value (value) files in  
sparse format, compatible with the  
Weka software.  
Concatenate information from  
HDR and SVM

```
@RELATION ~/Users/marcou/Desktop/Tutorial/CViter1Fold1/test.sdf"
@ATTRIBUTE "C" NUMERIC
@ATTRIBUTE "N" NUMERIC
@ATTRIBUTE "O" NUMERIC
@ATTRIBUTE "(C-C),xC" NUMERIC
@ATTRIBUTE "(C-C-N),xC" NUMERIC
@ATTRIBUTE "(C-C-N-C),(C-C-N-C),xC" NUMERIC
@ATTRIBUTE "(C-C),(C-N),xC" NUMERIC
@ATTRIBUTE "(C-N-C),(C-N-C),xC" NUMERIC
@ATTRIBUTE "(C-N-C-C),(C-N-C-N),(C-N-C=O),xC" NUMERIC
@ATTRIBUTE "(N-C),(N-C),(N-C),xN" NUMERIC
@ATTRIBUTE "(N-C-C),(N-C-C),(N-C-N),(N-C=O),xN" NUMERIC
@ATTRIBUTE "(N-C-N-C),xN" NUMERIC
@ATTRIBUTE "(C-N),(C-N),(C=O),xC" NUMERIC
0 8.05 1:22 2:3 3:2 7:1 10:1 13:1 14:1 16:1 17:1 18:1 19:1 39:1 42:4 45:7 46:1 48:1 62:2 63:
0 6.6 1:24 2:1 3:6 16:2 24:1 27:1 30:2 31:2 39:4 42:2 45:8 46:2 48:3 108:2 114:1 115:2 117:2
0 6.9 1:14 2:4 3:1 4:1 8:1 13:1 16:1 17:1 19:2 45:5 50:2 52:1 55:1 62:1 66:1 78:1 86:1 88:1
0 8.0 1:11 2:2 3:1 24:1 39:1 42:1 45:3 50:1 52:1 53:1 55:1 58:1 62:1 63:1 64:1 65:1 66:1 67:
0 8.65 1:20 2:3 3:1 4:2 5:2 6:2 7:2 8:2 10:1 16:1 24:1 27:1 28:1 30:1 31:1 32:1 33:1 36:1 39
0 8.2 1:19 2:2 4:1 24:2 27:1 32:1 33:1 34:1 36:2 37:1 39:3 42:2 43:1 44:1 45:3 46:1 47:1 48:
0 7.0 1:22 2:4 3:2 4:2 7:1 10:1 13:1 14:1 16:1 17:1 18:1 19:1 39:1 42:3 45:6 46:1 62:2 65:2
0 6.45 1:22 2:3 3:2 4:1 7:2 16:1 19:1 24:3 27:1 28:1 36:1 39:5 42:1 45:4 48:2 50:1 52:1 55:1
0 7.48 1:32 2:5 3:5 4:4 7:1 10:2 16:3 19:1 21:2 24:1 27:1 28:1 30:1 31:1 32:1 33:1 34:1 35:1
0 7.4 1:25 2:2 3:1 7:2 10:1 24:3 27:1 28:1 36:1 42:2 45:14 46:4 48:8 74:1 88:1 91:1 108:4 11
0 8.95 1:23 2:2 3:3 4:4 16:1 21:1 24:1 27:1 28:1 30:1 31:1 32:1 33:1 34:1 35:1 36:2 37:1 39:
0 7.5 1:23 2:2 3:1 24:3 27:1 28:1 39:4 42:2 45:8 46:2 48:4 50:1 52:1 53:1 55:1 58:1 62:1 63:
0 9.0 1:18 2:4 3:2 4:1 7:2 10:1 24:3 27:1 28:1 42:2 45:7 46:3 48:3 88:2 91:1 108:2 110:1 140
0 6.0 1:22 2:1 3:3 24:3 27:1 28:1 36:1 39:3 42:2 45:7 46:3 48:2 68:1 74:2 107:2 108:2 109:1
0 6.9 1:17 2:1 3:2 24:1 27:1 28:1 30:1 31:1 32:1 33:1 39:2 42:5 45:5 46:3 48:1 54:1 60:1 108
0 7.1 1:19 2:5 3:1 7:4 10:2 13:1 16:1 17:1 24:2 27:1 39:1 45:4 48:2 110:1 166:1 232:1 237:1
0 8.1 1:11 2:2 3:1 4:1 33:1 39:1 42:1 45:3 48:1 50:1 52:1 53:1 55:1 58:1 60:1 62:1 63:1 64:1
0 6.9 1:14 2:2 3:1 24:2 39:1 42:1 45:3 50:1 52:1 53:1 55:1 62:1 63:1 65:1 66:1 67:1 70:1 84:6
0 5.3 1:16 2:1 3:1 24:1 39:1 42:2 45:9 46:4 48:3 68:3 108:2 110:2 111:3 115:2 117:1 118:1 123:
0 6.6 1:15 2:1 3:2 4:1 7:1 16:1 19:1 39:1 42:1 45:6 46:1 48:1 55:2 99:1 102:1 115:1 117:1 114:
0 8.95 1:27 2:3 3:1 4:1 16:1 24:3 27:1 28:1 39:3 42:3 45:8 46:4 48:2 62:1 63:1 65:1 68:2 78:1
0 22, 1 3, 2 2, 3 1, 6 2, 15 1, 18 1, 23 3, 26 1, 27 1, 35 1, 38 5, 41 1, 44 4, 47 2, 49
0 32, 1 5, 2 5, 3 4, 6 1, 9 2, 15 3, 18 1, 20 2, 23 1, 26 1, 27 1, 29 1, 30 1, 31 1, 32 1
0 25, 1 2, 2 1, 6 2, 9 1, 23 3, 26 1, 27 1, 35 1, 41 2, 44 14, 45 4, 47 8, 73 1, 87 1, 90
0 23, 1 2, 2 3, 3 4, 15 1, 20 1, 23 1, 26 1, 27 1, 29 1, 30 1, 31 1, 32 1, 33 1, 34 1, 35
0 23, 1 2, 2 1, 23 3, 26 1, 27 1, 38 4, 41 2, 44 8, 45 2, 47 4, 49 1, 51 1, 52 1, 54 1, 5
0 18, 1 4, 2 2, 3 1, 6 2, 9 1, 23 3, 26 1, 27 1, 41 2, 44 7, 45 3, 47 3, 87 2, 90 1, 107
0 22, 1 1, 2 3, 23 3, 26 1, 27 1, 35 1, 38 3, 41 2, 44 7, 45 3, 47 2, 67 1, 73 2, 106 2,
0 17, 1 1, 2 2, 23 1, 26 1, 27 1, 29 1, 30 1, 31 1, 32 1, 38 2, 41 5, 44 5, 45 3, 47 1, 5
0 19, 1 5, 2 1, 6 4, 9 2, 12 1, 15 1, 16 1, 23 2, 26 1, 38 1, 44 4, 47 2, 109 1, 165 1, 2
0 11, 1 2, 2 1, 3 1, 32 1, 38 1, 41 1, 44 3, 47 1, 49 1, 51 1, 52 1, 54 1, 57 1, 59 1, 61
0 14, 1 2, 2 1, 23 2, 38 1, 41 1, 44 3, 49 1, 51 1, 52 1, 54 1, 61 1, 62 1, 64 1, 65 1, 6
0 16, 1 1, 2 1, 23 1, 38 1, 41 2, 44 9, 45 4, 47 3, 67 3, 107 2, 109 2, 110 3, 114 2, 11
0 15, 1 1, 2 2, 3 1, 6 1, 15 1, 18 1, 38 1, 41 1, 44 6, 45 1, 47 1, 54 2, 98 1, 101 1, 11
0 27, 1 3, 2 1, 3 1, 15 1, 23 3, 26 1, 27 1, 38 3, 41 3, 44 8, 45 4, 47 2, 61 1, 62 1, 64
```

# Summary

- **Division of the dataset into 4 sets of training and test structural files**
- **Calculation of the ISIDA Substructural Molecular Fragments descriptors**
  - ✓ Atom Count
  - ✓ Atom Centered fragment including from 2 to 4 atoms
  - ✓ Atoms and bonds standard annotation
  - ✓ Formal Charge

# Exercise 3

- **Goal:**

- ✓ Setup Gaussian Processes models using only the training sets.

- **Software:**

- ✓ Weka/Experimenter

- **File**

- ✓ CV/trainIteriFoldj.sdf and CV/train\_IteriFoldj\_\*.arff

- **Output**

- ✓ ExtCVtrain.arff
- ✓ ExtCVtrain.exp



# Introduction to Gaussian Processes

## ■ Goal:

✓ Model a probability distribution of the property.

- $Y^{train}, Y^{test}$  are random vectors representing properties
- $X^{train}, X^{test}$  are the molecular descriptors matrices

$$\begin{bmatrix} Y^{train} \\ Y^{test} \end{bmatrix} = N \left( 0, \begin{bmatrix} K(X^{train}, X^{train}) + \mathbf{1}\sigma^2 & K(X^{train}, X^{test}) \\ K(X^{test}, X^{train}) & K(X^{test}, X^{test}) \end{bmatrix} \right)$$

## ■ Parameters of the method

✓  $K(.,.)$  the kernel. In this tutorial, the linear kernel is used.

✓  $\sigma^2$  the noise level. Closely related to the experimental noise on the property.

# Inference of Gaussian Processes

## Two quantities are inferred for a test set

✓ The mean vector

$$\langle Y^{test} \rangle = K(X^{test}, X^{train}) [K(X^{train}, X^{train}) + \mathbf{1}\sigma^2]^{-1} Y^{train}$$

✓ The covariance

$$\sigma_{Y^{test}}^2 = K(X^{test}, X^{test}) -$$

$$K(X^{test}, X^{train}) [K(X^{train}, X^{train}) + \mathbf{1}\sigma^2]^{-1} K(X^{train}, X^{test})$$



(•) Data points

(—) Mean vector

(—) +/- two variances

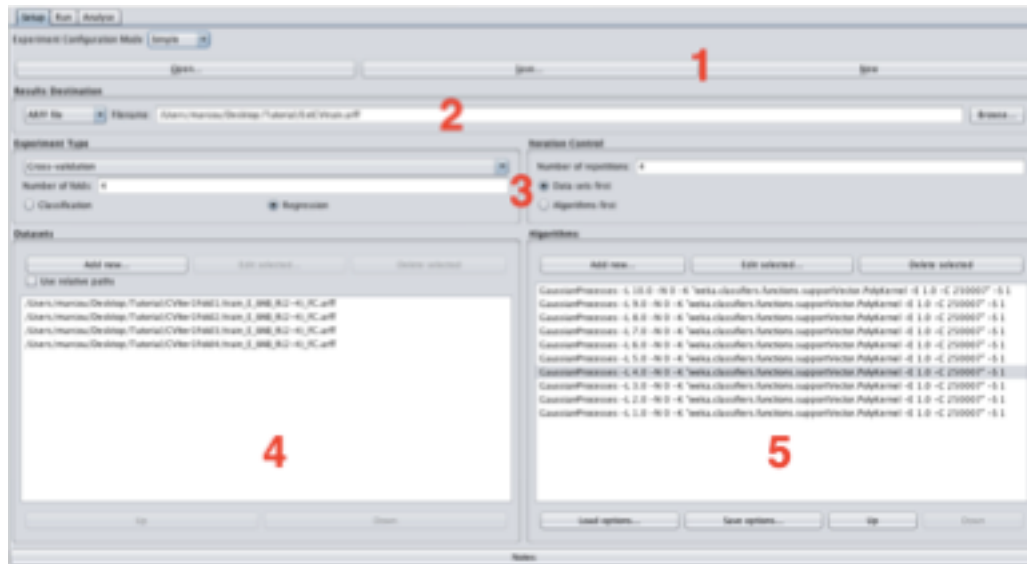
# Weka/Experimenter

The screenshot shows the Weka Experimenter interface with five red numbers highlighting specific features:

- 1**: Points to the **Save...** button in the top toolbar.
- 2**: Points to the **Filename** input field in the **Results Destination** section, containing the path `/Users/marcou/Desktop/Tutorial/ExpCVtrain.arff`.
- 3**: Points to the **Iteration Control** section, where **Data sets first** is selected.
- 4**: Points to the **Datasets** list, which contains four entries for cross-validation folds.
- 5**: Points to the **Algorithms** list, which contains ten entries for `GaussianProcesses` with varying parameters.

The interface includes sections for **Experiment Configuration Mode** (Simple), **Results Destination**, **Experiment Type** (Cross-validation, Regression), **Iteration Control** (Number of repetitions: 4), **Datasets**, and **Algorithms**. The **Notes** section is visible at the bottom.

# Weka/Experimenter



- ✓ Click New.
- ✓ Results destination: “ExtCVtrain.arff”

- ✓ 4-fold cross validation repeated 4 times.
- ✓ Experiment type: Regression
- ✓ Add the training files to the Datasets
- ✓ Add the ZeroR method to the Algorithms
- ✓ Add Gaussian Processes method to the Algorithms

# Setup Gaussian Processes

Choose weka.classifiers.functions.GaussianProcesses

About

Implements Gaussian processes for regression without hyperparameter-tuning. More Capabilities

batchSize 100

debug False

doNotCheckCapabilities False

filterType No normalization/standardization

kernel Choose PolyKernel -E 1.0 -C 250007

noise 1.0

numDecimalPlaces 2

seed 1

Open... Save... OK Cancel

Add Gaussian Processes:

- no data transformations
- Noise level in the range [1..10]

# Finalize the experiment

- Click the button *Save...*
  - ✓ Save you setup as *ExtCVtrain.exp*
- Click the *Run* tab, then *Start*



- Click the *Analyse* tab



# Analyze the experiment on training data

- Click the *Experiment* button
- Set the *Comparison field* to *Relative Absolute Error*
- Click the *Perform test* button

Dataset	(1) rules.Zero	(2) funct	(3) funct	(4) funct	(5) funct	(6) funct	(7) funct	(8) funct	(9) funct	(10) funct	(11) funct
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



# Summary

- **The optimum noise level is deduced only from the internal cross-validation on training sets.**
  - ✓ Optimal value about 4
- **Different performances on different training sets**
  - ✓ Must be statistically demonstrated
- **The test sets are fully independent.**
  - ✓ The setup of the model building is complete without the help of the test sets



# Exercise 4

- **Goal:**

- ✓ External validation of Gaussian Processes models.

- **Software:**

- ✓ Weka/Experimenter

- **File**

- ✓ CV/trainIteriFoldj.sdf and CV/trainIteriFoldj\_\*.arff
- ✓ CV/testIteriFoldj.sdf and CV/testIteriFoldj\_\*.arff

- **Output**

- ✓ ExtCVtrain.arff
- ✓ ExtCVtrain.exp

# Weka/Experimenter Advanced mode

Setup Run Analyse

Experiment Configuration Mode **Advanced**

Open... Save... **1** New

**Destination**

Choose InstancesResultListener -O ExpCVtest.arff **2**

**Result generator**

Choose ExplicitTestsetResultProducer -R -O splitEvaluatorOut.zip -dir /Users/marcou/Desktop/Tutorial/testsets -suffix \_test.arff -W weka.experiment.RegressionSplitEvaluator -- - **3**

**Runs** From: 1 To: 1

**Distribute experiment**

Hosts

By data set  By run  By property

**Generator properties**

Disabled Select property...

Can't edit

**Iteration control**

Data sets first  Custom generator first

**Datasets**

Add new... Edit selected... Delete selected

Use relative paths

/Users/marcou/Desktop/Tutorial/CVIter1Fold1/train\_E\_IAB\_R(2-4)\_FC.arff **4**

/Users/marcou/Desktop/Tutorial/CVIter1Fold2/train\_E\_IAB\_R(2-4)\_FC.arff

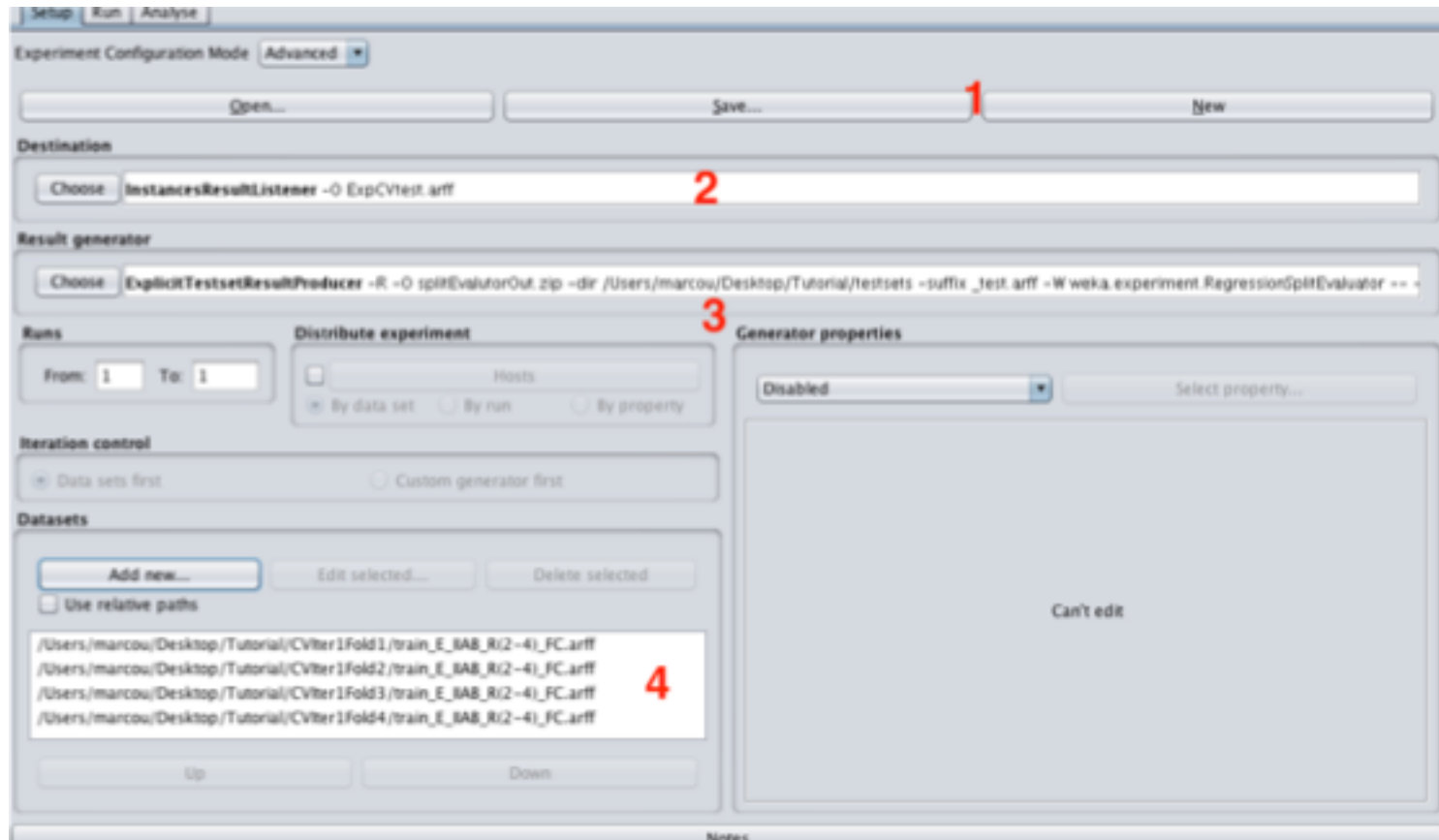
/Users/marcou/Desktop/Tutorial/CVIter1Fold3/train\_E\_IAB\_R(2-4)\_FC.arff

/Users/marcou/Desktop/Tutorial/CVIter1Fold4/train\_E\_IAB\_R(2-4)\_FC.arff

Up Down

# Weka/Experimenter Advanced mode

- Set the *Destination* to *ExpCVtest.arff*
- Set the *Runs* from 1 to 1
- Check or set the *Datasets*



# Setup the Results Generator

weka.experiment.ExplicitTestsetResultProducer

**About**

Loads the external test set and calls the appropriate SplitEvaluator to generate some results. More

outputFile

randomizeData

rawOutput

relationFind

relationReplace

splitEvaluator  **RegressionSplitEvaluator -W weka.classifiers.**

testsetDir

testsetPrefix

testsetSuffix

- Set the value of *RelationFind* to *(.\*train|\.sdf.\*)*
- Set the value of *testsetPrefix* to *test*
- Set the value of *testsetSuffix* to *\_E\_IIAB(2-4)\_FC.arff*
- Set the *testsetDir* to *CV*
- Click the *RegressionSplitEvaluator*

# RegressionSplitEvaluator: GP with optimum noise level

weka.classifiers.functions.GaussianProcesses

**About**

weka.experiment.Regresion/

**About**

A SplitEvaluator that pro...

More

Capabilities

More

More

classifier Cho

noSizeDetermination False

-C 250007 -S 1

Open...

batchSize 100

debug False

doNotCheckCapabilities False

filterType No normalization/standardization

kernel Choose PolyKernel -E 1.0 -C 250007

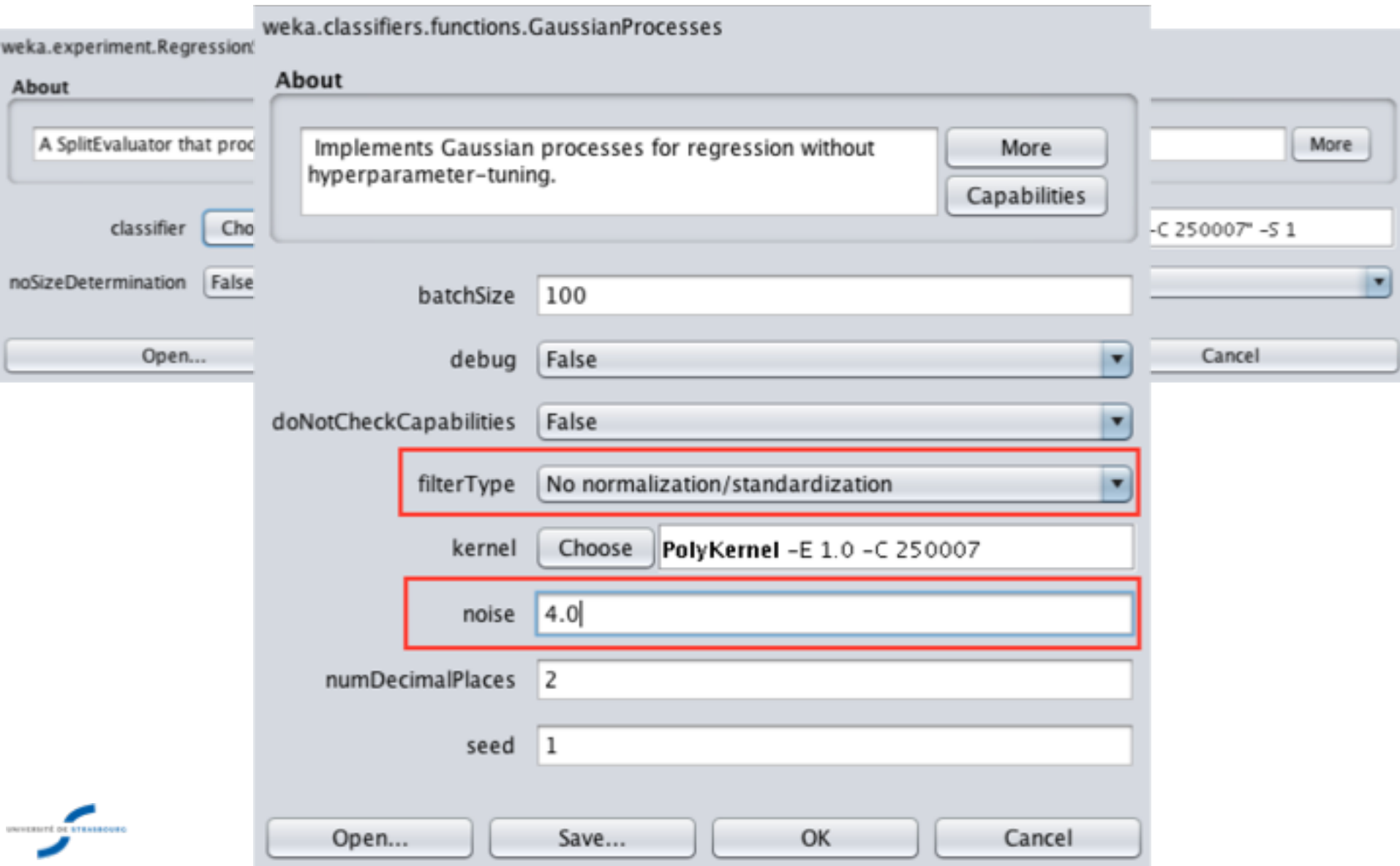
noise 4.0

numDecimalPlaces 2

seed 1

Cancel

Open... Save... OK Cancel



# Finalize the experiment

- Click the button *Save...*
  - ✓ Save you setup as *ExtCVtrain.exp*
- Click the *Run* tab, then *Start*



- Click the *Analyse* tab



# Analyze the experiment on training data

- Click the *Experiment* button
- Set the *Comparison field* to *Relative Absolute Error*
- Click the *Perform test* button

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

```
01:24:54 - Available resultsets
01:24:55 - Root_mean_squared_error - /users/marco
01:25:42 - Available resultsets
01:25:45 - Root_mean_squared_error - functions.Gauss
01:25:58 - Relative_absolute_error - functions.Gauss
01:26:35 - Relative_absolute_error - functions.Gauss
01:26:54 - Relative_absolute_error - functions.Gauss
01:28:14 - Relative_absolute_error - functions.Gauss
```

```
(7) functions.GaussianProcesses '-L 4.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(8) functions.GaussianProcesses '-L 3.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(9) functions.GaussianProcesses '-L 2.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
(10) functions.GaussianProcesses '-L 1.0 -N 2 -K \"functions.supportVector.PolyKernel -E 1.0 -C 250007\" -S 1' -8620066949967678545
```

# Summary

- **The optimum noise level is deduced only from the internal cross-validation on training sets.**
  - ✓ Optimal value about 4
- **The configuration is used to validate the models on the test sets**
  - ✓ Relative absolute error about 80%
- **Performances are dependent of the cross-validation fold**
  - ✓ Are there outliers?



# Exercise 4

- **Goal:**
  - ✓ Outlier identification
- **Software:**
  - ✓ Weka/Explorer
  - ✓ xModelAnalyzerR
  - ✓ InstantJChem
- **Files**
  - ✓ ARFF files
- **Output**
  - ✓ Log files of Weka/Explorer
  - ✓ New SDF without an outlier.

# Weka/Explorer

Load the file IUPHAR\_5HT2B\_E\_IIAB\_R(2-4)\_FC.arff

The screenshot shows the Weka Explorer interface. At the top, there are tabs for 'Preprocess', 'Classify', 'Cluster', 'Associate', 'Select attributes', and 'Visualize'. Below these are buttons for 'Open file...', 'Open URL...', 'Open DB...', 'Generate...', 'Undo', 'Edit...', and 'Save...'. The 'Open file...' button is highlighted with a red rectangle. Below the buttons is a 'Filter' section with a 'Choose' button and a text field containing 'None', and an 'Apply' button. The 'Current relation' section shows 'Relation: /Users/marcou/Desk...' and 'Instances: 88'. The 'Attributes' section has buttons for 'All', 'None', 'Invert', and 'Pattern', and a list of attributes with checkboxes. The 'Selected attribute' section shows 'Name: class', 'Missing: 0 (0%)', 'Distinct: 49', and 'Type: Numeric'. Below this is a table of statistics for the 'class' attribute. The 'Class: class (Num)' dropdown is set to 'class (Num)', and there is a 'Visualize All' button. A histogram is displayed below the dropdown, showing the distribution of the 'class' attribute. The 'Status' section at the bottom shows 'OK' and a 'Log' button.

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Choose None Apply

Current relation

Relation: /Users/marcou/Desk... Attributes: 1436  
Instances: 88 Sum of weights: 88

Attributes

All None Invert Pattern

No.	Name
1428	(O=C*N),(O=C*O),xO
1429	(O=C*N*N),(O=C*N-C),(O=C*O*C),xO
1430	(O*C),(O*C),xO
1431	(O*C*N),(O*C*N),(O*C-C),(O*C=O),xO
1432	(O*C*N-C),(O*C-C*C),(O*C-C*C),xO
1433	(C-C-C),(C-C-C),(C-N*C),(C-N*N),xC
1434	(C-C-C-N&FC+1&),(C-C-C-N&FC+1&),(C-...
1435	(C-C-C-C),(C-C-N&FC+1&-C),(C-C-N&FC+...
1436	class

Remove

Selected attribute

Name: class  
Missing: 0 (0%)  
Distinct: 49  
Type: Numeric  
Unique: 24 (27%)

Statistic	Value
Minimum	5.2
Maximum	10.05
Mean	7.339
StdDev	1.156

Class: class (Num) Visualize All

5.2 7.63 10.05

Status

OK Log x 0

# Setup a Gaussian Processes model

- Setup a (weka.classifiers.functions.GaussianProcesses
- Select the
- Click the

weka.classifiers.functions.GaussianProcesses

About

Implements Gaussian processes for regression without hyperparameter-tuning.

More

Capabilities

Choose

Test option

Use tri

Suppli

Cross-

Percen

batchSize 100

debug False

doNotCheckCapabilities False

filterType No normalization/standardization

kernel Choose PolyKernel -E 1.0 -C 250007

noise 4.0

numDecimalPlaces 2

seed 1

Start

Result list

16:19:15

16:20:38

16:21:44

16:22:06

Open... Save... OK Cancel

# Weka/Explorer Gaussian Processes

## Cross-Validation results

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

```
==== Classifier model (from training set) ====
```

Gaussian Processes

```
Kernel used:  
  Linear Kernel:  $K(x,y) = \langle x,y \rangle$ 
```

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

- Experiment summary
- Description of the model
- Cross-validation statistics

# Setup a Gaussian Processes model

- Setup a Gaussian Processes model if needed
- Select the *Supplied test set* option and set the training file as test. This produces fitting results.
- Click the *More options...* button.

The screenshot shows the Weka Classifier interface. The 'Classifier' tab is selected, and the 'GaussianProcesses' model is chosen. The 'Test options' section shows 'Supplied test set' selected. The 'Classifier output' window displays the following results:

```
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

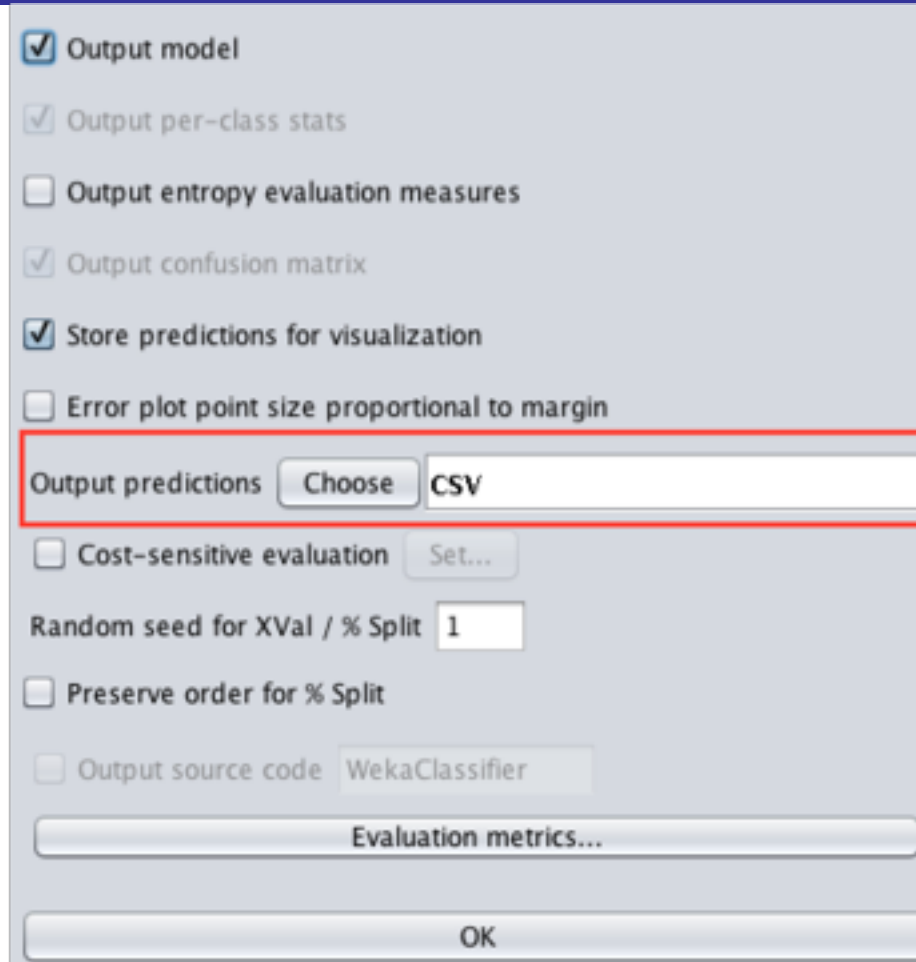
=== Summary ===
Correlation coefficient           0.979
Mean absolute error              0.2359
Root mean squared error          0.2917
Relative absolute error          23.9013 %
Root relative squared error      25.3818 %
Total Number of Instances       88
```

The 'Result list (right-click for options)' shows the following entries:

- 16:19:15 - functions.GaussianPro
- 16:20:38 - functions.GaussianPro
- 16:21:44 - functions.GaussianPro
- 16:22:06 - functions.GaussianPro

The 'Status' bar at the bottom shows 'OK' and 'Log' buttons.

# Weka/Explorer classifier options



Output model

Output per-class stats

Output entropy evaluation measures

Output confusion matrix

Store predictions for visualization

Error plot point size proportional to margin

Output predictions  CSV

Cost-sensitive evaluation

Random seed for XVal / % Split

Preserve order for % Split

Output source code

- Set the *Output predictions* to *CSV*.
- Click *OK* then click the button *Start*.

# Weka/Explorer Gaussian Processes

## Fitting results

```
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

- Estimation for each instance

- ✓ Weka/Explorer does not provide the covariance matrix.

- Fit performances.

- ✓ As they are excellent, the outliers become visible
- ✓ Outliers *do not* fit.

# Save Weka result buffer

The screenshot shows the Weka Classifier window. The 'Classifier' dropdown is set to 'GaussianProcesses'. The 'Test options' section has 'Supplied test set' selected. The 'Classifier output' table contains the following data:

Line	Value
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

A right-click context menu is open over the output table, with the 'Save result buffer' option highlighted. A red arrow points from the text 'Right click on the model's line' to the menu, and another red arrow points from the text 'Save your file as GP\_Fit\_all.out' to the 'Save result buffer' option.

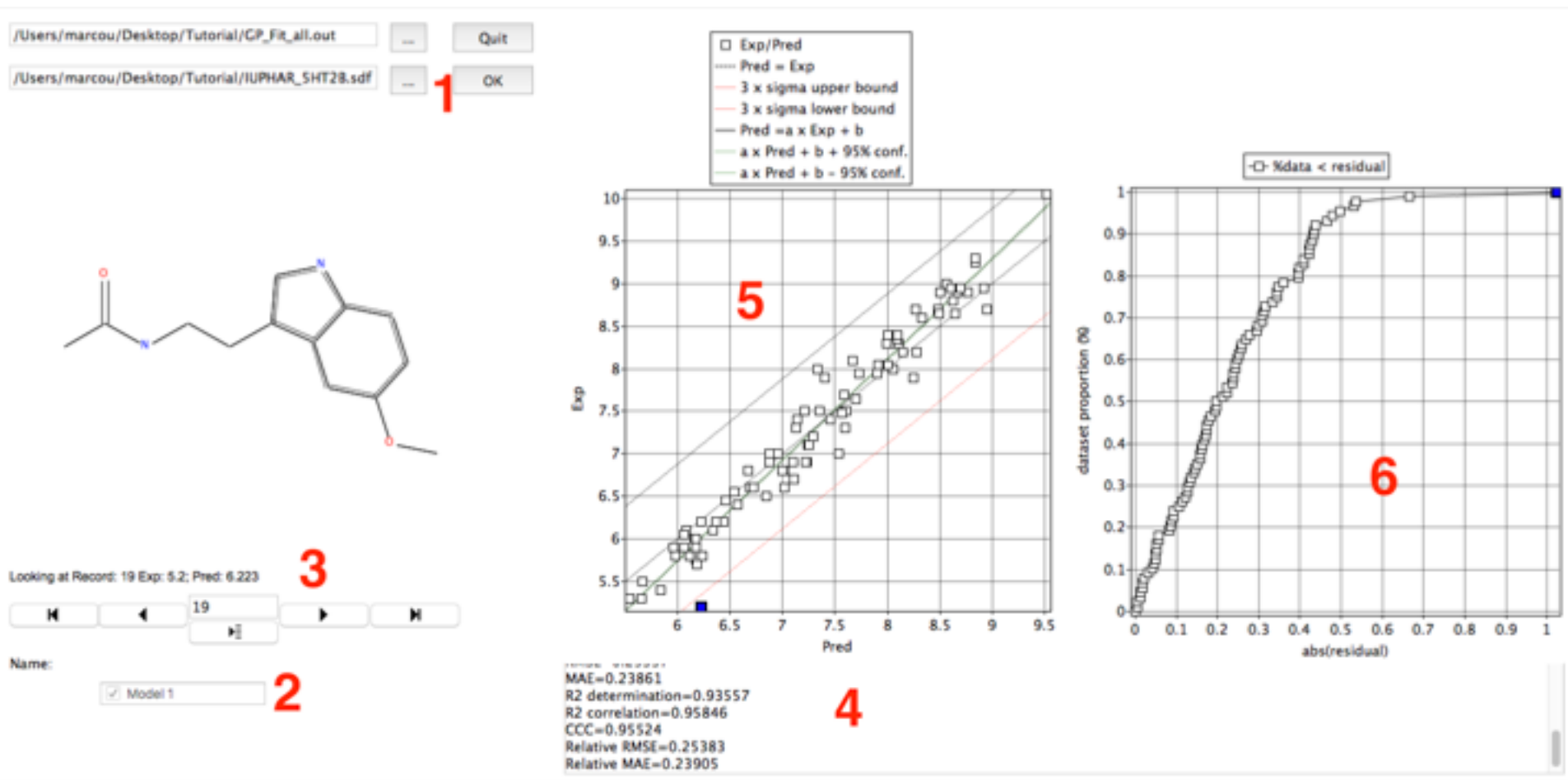
- Right click on the model's line

- Save your file as *GP\_Fit\_all.out*



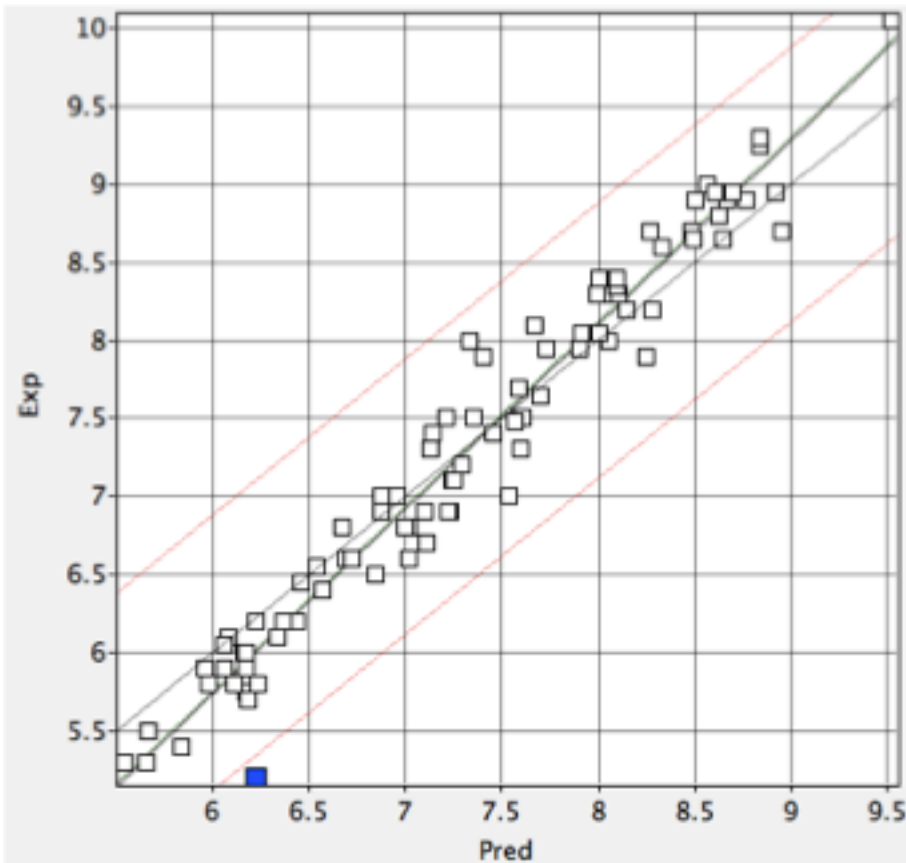
# xMolDelAnalyzerR

- Open the file GP\_Fit\_all.out and the file IUPHAR\_5HT2B.sdf.
- Click the OK button.



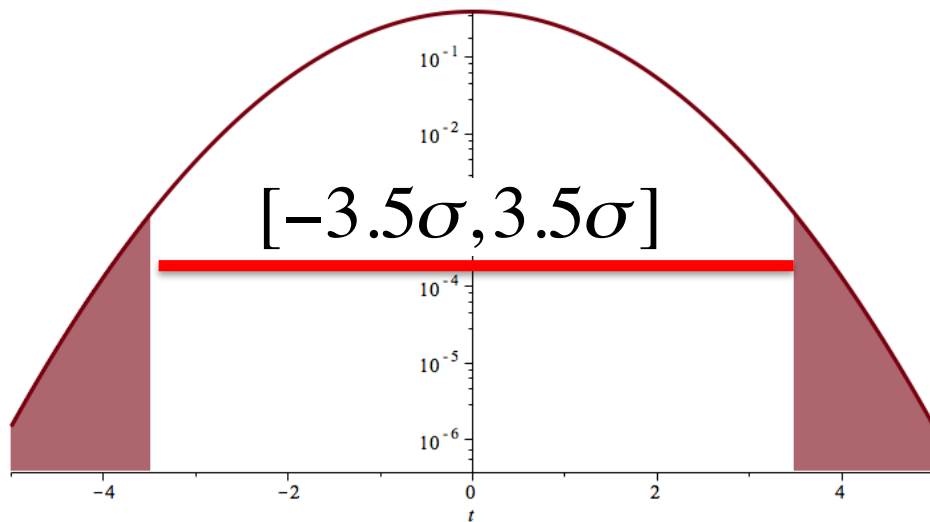
# Outlier detection

## Example of a single step procedure



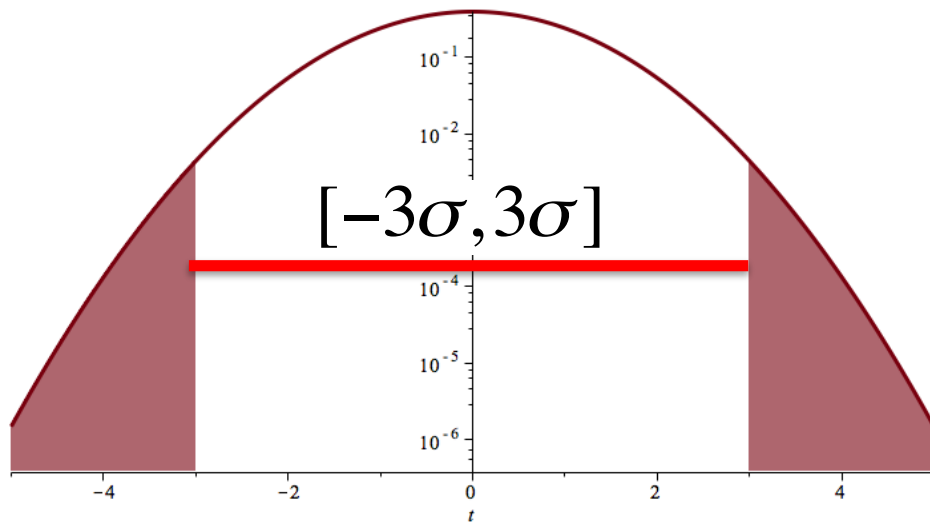
- All points outside the  $[-3\sigma, 3\sigma]$  range.
  - ✓ Assumption: data follow a normal distribution.
  - ✓ High impact of outliers on the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) values.

# Interval width and risk in outlier detection



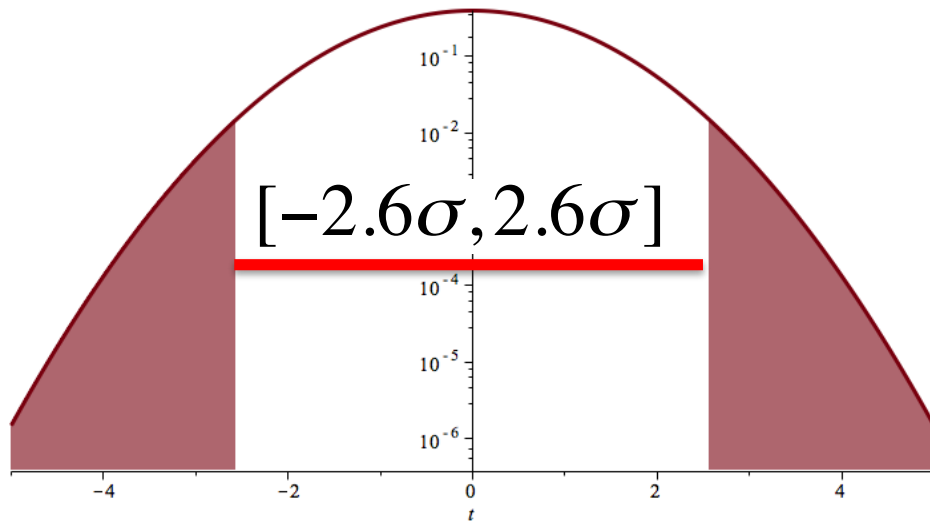
Risk(%)	u (sigma)
<b>0.05</b>	<b>3.5</b>

# Interval width and risk in outlier detection



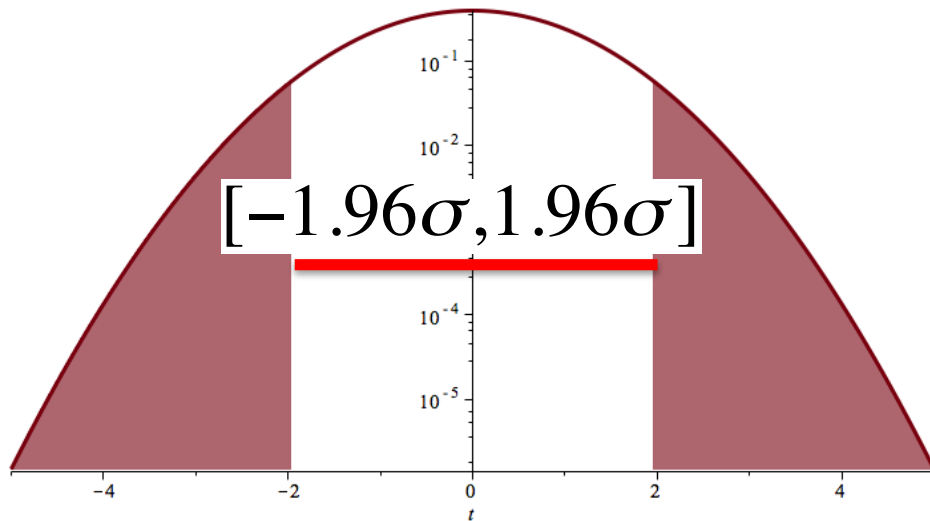
Risk(%)	u (sigma)
0.05	3.5
<b>0.25</b>	<b>3</b>

# Interval width and risk in outlier detection



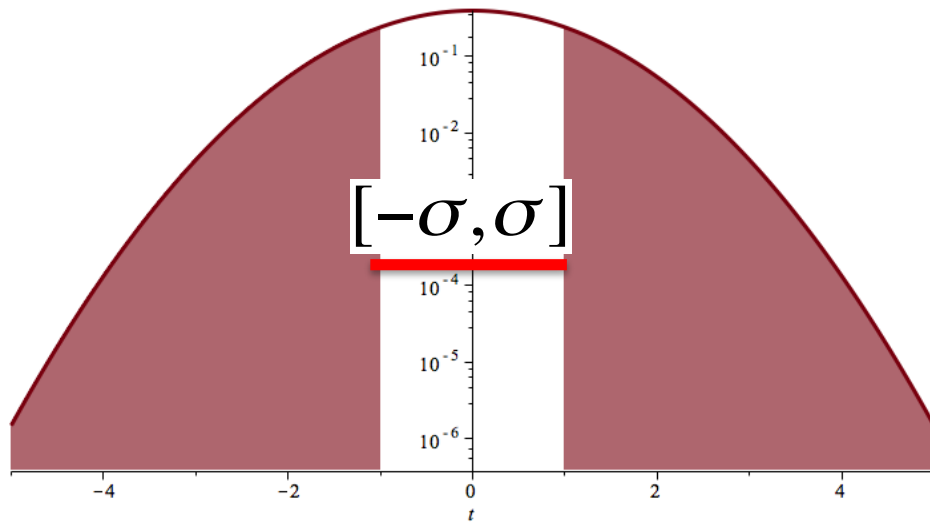
Risk(%)	u (sigma)
0.05	3.5
0.25	3
<b>1.00</b>	<b>2.6</b>

# Interval width and risk in outlier detection



Risk(%)	u (sigma)
0.05	3.5
0.25	3
1.00	2.6
<b>5.00</b>	<b>1.96</b>

# Interval width and risk in outlier detection



Risk(%)	u (sigma)
0.05	3.5
0.25	3
1.00	2.6
5.00	1.96
<b>32.00</b>	<b>1.00</b>

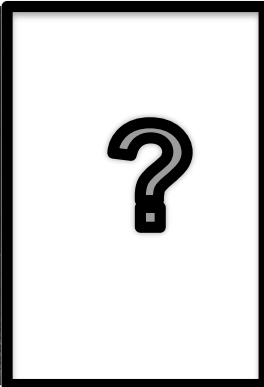
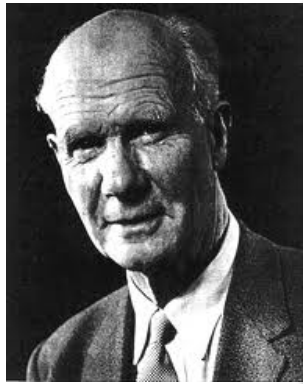
# Outlier detection

## Example of sequential procedure



WR Thompson (?)

Formulation of the test in 1935



ES Pearson  
C Chandra Sekar

Domain of validity of the test, 1936



FE Grubbs

Tables of critical values, 1950

### ■ The Grubbs test.

✓ Compute decision variables:

$$G_1 = \frac{\langle r \rangle - r_1}{s}$$

$$G_N = \frac{r_N - \langle r \rangle}{s}$$

✓ Compute critical value:

$$G_c = \frac{N-1}{N} \sqrt{\frac{t^2}{N-2+t^2}}$$

$t, \alpha/2N$   
fractile of student distribution with  $N-2$  degrees of freedom

✓ Take a decision:

$$G_1 > G_c \Rightarrow r_1, \text{ outlier}$$

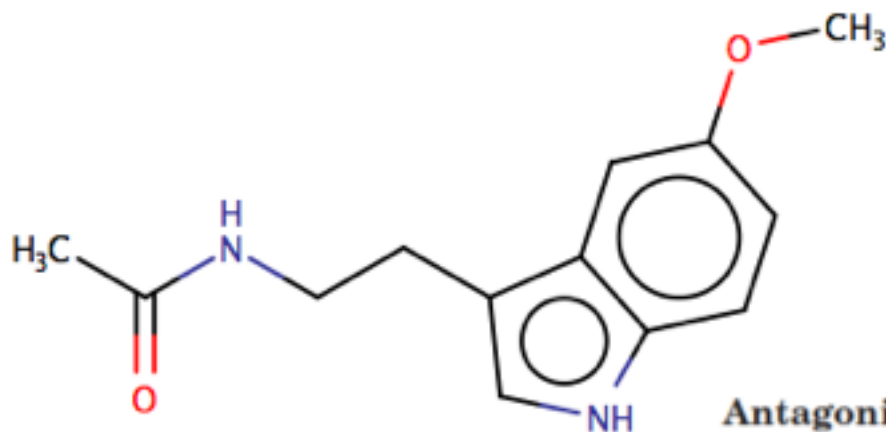
$$G_N > G_c \Rightarrow r_N, \text{ outlier}$$



# Outlier identification

## ■ Outlier, compound 33

- ✓ Melatonin
- ✓ PubID 12750432



0022-3859/03/26(10)-954-964E\$05  
The Journal of Pharmacology and Experimental Therapeutics  
Copyright © 2003 by The American Society for Pharmacology and Experimental Therapeutics  
JPEP 26(10)-954, 2003

Vol. 266, No. 10  
October 2003  
Printed in U.S.A.

The Novel Melatonin Agonist Agomelatine (S20098) Is an Antagonist at 5-Hydroxytryptamine<sub>2C</sub> Receptors, Blockade of Which Enhances the Activity of Frontocortical Dopaminergic and Adrenergic Pathways

M. J. MILLAN, A. GOBERT, F. LEJEUNE, A. DEKEYNE, A. NEWMAN-TANCREDI, V. PASTEAU, J.-M. RIVET, and D. CUSSAC

Department of Psychopharmacology, Institut de Recherches Servier, Croissy/Seine, France

Received March 18, 2003; accepted April 30, 2003

TABLE 1

Binding affinities of agomelatine compared to melatonin at 5-HT<sub>2</sub> receptor subtypes

Data are mean ± S.E.M. of pK<sub>i</sub> values derived from at least three independent determinations, each of which was performed in triplicate.

Drug	h5-HT <sub>2C</sub>	h5-HT <sub>2B</sub>	h5-HT <sub>2A</sub>	α5-HT <sub>2A</sub>	α5-HT <sub>2C</sub>
Agomelatine	6.15 ± 0.04	6.59 ± 0.07	5.35 ± 0.06	<5.0	6.39 ± 0.02
Melatonin	<5.0	5.24 ± 0.06	<5.0	<5.0	<5.0

h, human; r, rat; p, porcine.

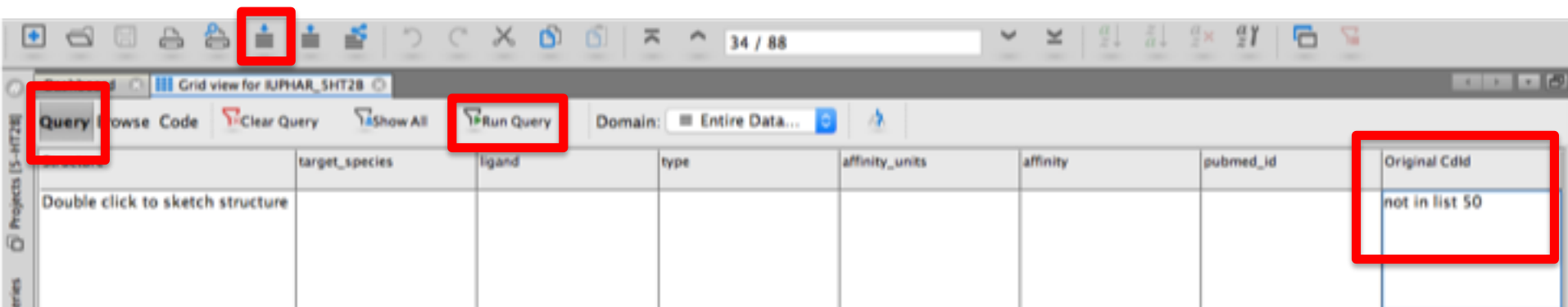
**Antagonist Properties of Agomelatine and Melatonin at h5-HT<sub>2B</sub> Receptors: [<sup>3</sup>H]PI Depletion (Fig. 4; Table 2).** Agomelatine failed to elicit [<sup>3</sup>H]PI depletion alone and concentration dependently blocked the action of 5-HT with a pK<sub>B</sub> value of 6.6 corresponding well to its pK<sub>i</sub> value (6.6) at these sites. Melatonin likewise did not enhance [<sup>3</sup>H]PI depletion and partially attenuated the action of 5-HT, although only ~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.

# Exercise 5

- **Goal:**
  - ✓ Outlier identification
- **Software:**
  - ✓ Weka/Explorer
  - ✓ xModelAnalyzerR
  - ✓ InstantJChem
- **Files**
  - ✓ ARFF files
- **Output**
  - ✓ Log files of Weka/Explorer
  - ✓ New SDF without an outlier.

# Remove Melatonin from the training set

- In *InstantJChem* software, click the *Query* button.
- Set the query field Original Cdid to « not in list 50 » - the Original Cdid of Melatonin.
- Click the *Run Query* button.



The screenshot shows the InstantJChem software interface. The 'Query' button is highlighted with a red box. The 'Run Query' button is also highlighted with a red box. The 'Original Cdid' column in the table is highlighted with a red box, containing the text 'not in list 50'. The table has the following columns: target\_species, ligand, type, affinity\_units, affinity, pubmed\_id, and Original Cdid. The first row of the table contains the text 'Double click to sketch structure'.

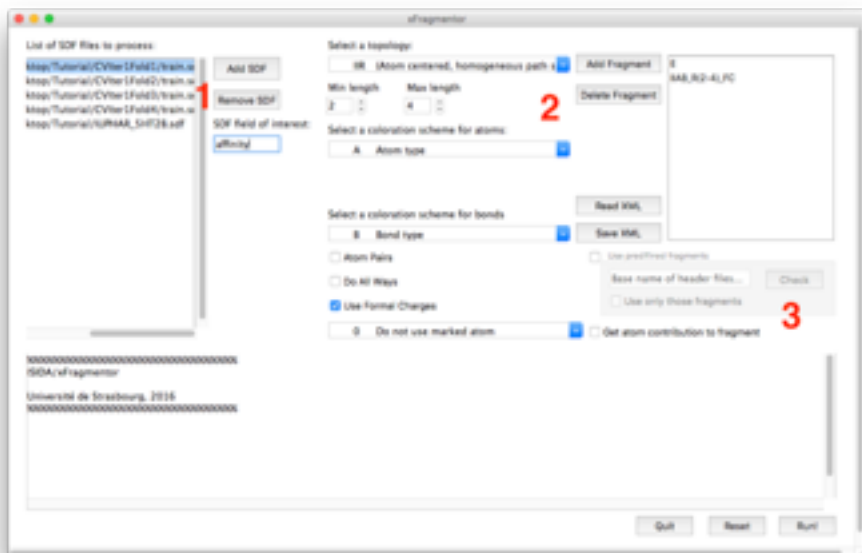
target_species	ligand	type	affinity_units	affinity	pubmed_id	Original Cdid
Double click to sketch structure						not in list 50

- Click the export button.
- Save the exported SDF file as IUPHAR\_5HT2B-33.sdf.

# Fragment the curated dataset

## ■ In the *xFragmentor* interface

- ✓ If needed, click the Read XML button and read the file *train\_E\_IIAB\_R(2-4)\_FC.xml*
- ✓ Remove all SDF files
- ✓ Add the file IUPHA\_5-HT2B-33.sdf
- ✓ Click the button *Run*.



# Load the curated dataset into the Weka/Explorer software

Load the file IUPHAR\_5HT2B-33\_E\_IIAB\_R(2-4)\_FC.arff

The screenshot shows the Weka Explorer software interface. The 'Open file...' button is highlighted with a red box. The interface includes a menu bar with options: Preprocess, Classify, Cluster, Associate, Select attributes, and Visualize. Below the menu bar are buttons for 'Open file...', 'Open URL...', 'Open DB...', 'Generate...', 'Undo', 'Edit...', and 'Save...'. The 'Filter' section has a 'Choose' button and a dropdown menu set to 'None', with an 'Apply' button. The 'Current relation' section shows 'Relation: /Users/marcou/Desk...' and 'Instances: 88'. The 'Attributes' section has buttons for 'All', 'None', 'Invert', and 'Pattern', and a list of attributes with checkboxes. The 'Selected attribute' section shows 'Name: class', 'Type: Numeric', and a table of statistics. The 'Class: class (Num)' dropdown is set to 'class (Num)', and a 'Visualize All' button is present. A histogram shows the distribution of the 'class' attribute with bars at 5.2, 7.63, and 10.05. The 'Status' section shows 'OK' and a 'Log' button.

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Choose None Apply

Current relation

Relation: /Users/marcou/Desk... Attributes: 1436  
Instances: 88 Sum of weights: 88

Attributes

All None Invert Pattern

No.	Name
1428	(O=C*N),(O=C*O),xO
1429	(O=C*N*N),(O=C*N-C),(O=C*O*C),xO
1430	(O*C),(O*C),xO
1431	(O*C*N),(O*C*N),(O*C-C),(O*C=O),xO
1432	(O*C*N-C),(O*C-C*C),(O*C-C*C),xO
1433	(C-C-C),(C-C-C),(C-N*C),(C-N*N),xC
1434	(C-C-C-N&FC+1&),(C-C-C-N&FC+1&),(C-...
1435	(C-C-C-C),(C-C-N&FC+1&-C),(C-C-N&FC+...
1436	class

Remove

Selected attribute

Name: class  
Missing: 0 (0%)  
Distinct: 49  
Type: Numeric  
Unique: 24 (27%)

Statistic	Value
Minimum	5.2
Maximum	10.05
Mean	7.339
StdDev	1.156

Class: class (Num) Visualize All

5.2 7.63 10.05

Status

OK Log x 0

# Re-fit the Gaussian Processes model

- Setup a Gaussian Processes model if needed
- Select the *Supplied test set* option and set the training file as test. This produces fitting results.
- Click the *More options...* button.

The screenshot shows the Weka Classifier interface. The 'Classifier' dropdown is set to 'GaussianProcesses' with the command line: `-L 4.0 -N 2 -K "weka.classifiers.functions.supportVector.PolyKernel" -E 1.0 -C 250007* -S 1`. Under 'Test options', 'Supplied test set' is selected. The 'More options...' button is highlighted. The 'Classifier output' pane shows the following results:

```
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
```

The 'Result list' shows four entries for 'functions.GaussianPro' at different times: 16:19:15, 16:20:38, 16:21:44, and 16:22:06. The 'Status' bar at the bottom shows 'OK' and a 'Log' button.

# Re-evaluate the Gaussian Processes model

- Setup a Gaussian Processes model with an optimal noise value (for instance 4)
- Select the *Cross-validation* option and set the *Folds* value to 4.
- Click the *Start* button

The screenshot displays the Weka GUI for the Gaussian Processes classifier. The top toolbar shows the command: `Choose GaussianProcesses -L 4.0 -N 2 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -S 1`. In the **Test options** section, the **Cross-validation** radio button is selected, and the **Folds** value is set to 4. The **Start** button is highlighted. The **Classifier output** window shows the following results:

```
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
```

The **Result list** (right-click for options) shows a list of recent runs:

- 16:19:15 - functions.GaussianPro
- 16:20:38 - functions.GaussianPro
- 16:21:44 - functions.GaussianPro
- 16:22:06 - functions.GaussianPro

# Gaussian Processes model on currated dataset

## Fit

## Cross-validation

=== 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

=== 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

Performances are (slightly) improved



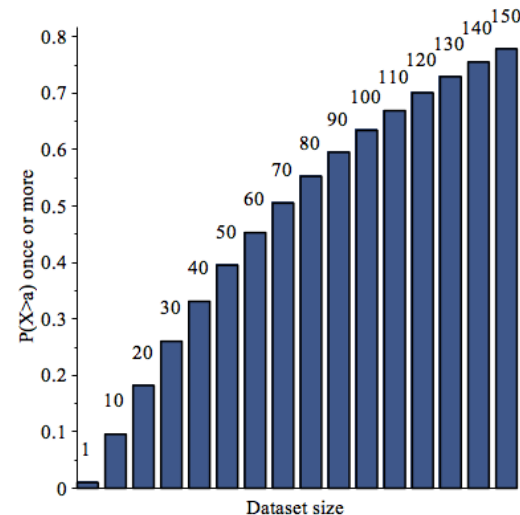
# Summary

## ■ Single step procedure

- ✓ Fast and simple
- ✓ False estimation of risk in outlier detection:
  - Distribution parameters are distorted by outliers
  - The risk estimation is valid for only one point.
- ✓ The bigger the dataset is, the greater the chances to interpret a standard instance as outlier (false positive)

$$P(X > a) = p$$

$$P(\{x_i \mid x_i > a\} \subset \text{Sample}) = \text{Binomial}(N, p)$$



# Summary

## ■ Sequential procedure

✓ Fastidious

- the whole procedure repeats after each identified outlier.

✓ Distribution parameters are distorted by the outlier

- Use robust statistics

resampling by the half-means method or by the smallest half-volume method

Egon WJ, Morgan SL, *Anal. Chem.*, 1998, **70**(11), pages 2372-2379

## ■ Identify multiple outliers

✓ Rank instances by outlier likeliness

Higher rank to instances that are consistently not fitted in consensus modeling

# Conclusion

- **Outlier detection may depend on data modeling**
  - ✓ Molecular Descriptors, Machine Learning method,...
  - ✓ Therefore, a consensus model-based approach may help (pick the outliers that are common to the several distinct models)
- **Use internal and external cross-validation to avoid overfitting.**
- **Exploit the fitted values (rather than cross-validated predictions) for outlier detection**
- **An “outlier” not confirmed as anomalous cannot be discarded**
  - ✓ but it may help understand the limitations of your model
  - ✓ It might be a “discovery”!

# Thanks



# Thank you

