Training the pKa Plugin

This manual gives you a walk-through on how to train the pK_a Plugin:

Introduction

If you think your experimental data can improve the accuracy of the pK_a calculation, you can take advantage of a supervised pK_a learning method that is built into the pK_a plugin. Special structural parts can have an effect on the pK_a values calculated by the built-in method, so your correction library based on your experimental data can help the pK_a plugin increase the prediction accuracy.

Inaccurately predicted ionization centers need to be identified and experimental data for them have to be collected in order to handle them. Since the learning algorithm is based on linear regression analysis, you need to collect as much experimental pK_a data as possible to get enough correlation. There are no hard-and-fast rules about the amount of data to be applied. If your are to create a local model only for a certain type of ionization centers, then it may be enough to collect a few representative structures. A robust model, however, requires as many diverse structures and pK_a values as possible.

The experimental data should be collected in an SD file. Then the training command has to be run in order to create a correction library. This will be stored on your local computer, in your user folder.

Finally, this correction library can be applied via MarvinSketch, Chemical Terms or cxcalc calculator functions command line tool.

Training steps

Preparing the input file

To create a training library a proper input file in SDF or MRV format should be prepared first. This file can be compiled using either Instant JChem or JChem for Excel.

The SD file should contain the following pieces of information:

- Structure of the molecule
- pK_a value 1 (field name: pKa1)
- ID of the atom which has the pKa1 value (field name: ID1). It can be viewed by checking the Atom number option in MarvinView (*View > Misc* menu).
- Additional fields of pK_a values are optional (recommended for handling multiprotic compounds). For example pK_a value 2 (pKa2), ID2, etc.
- Definition of only one pK_a value is enough to apply the training data, but more values in case of multiprotic compounds will enhance the reliability of the pK_a training.

A sample of a typical training set is shown in the picture (pKa_trainingset.sdf). ID1 is the index of the atom with the experimental pK_a1 value.

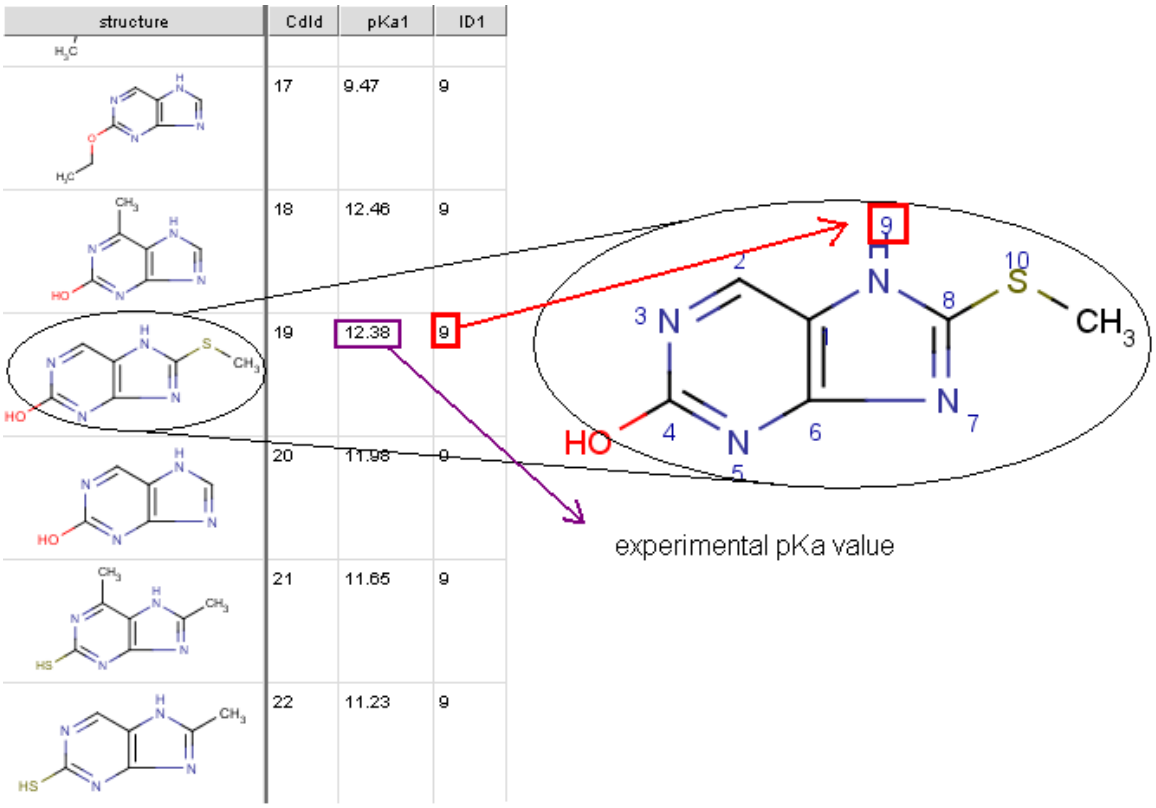


Fig. 1 Input for training library generation

Creating the training library

The training library can be created using the *cxtrain* command line tool from an input structural file:

```
cxtrain pka -i [library name] [training file]
```

Example:

```
cxtrain pka -i mypka mydata.sdf
```

Applying the training library

Once the training library is generated, it can be applied in different ChemAxon tools for training.

MarvinSketch

To apply the pre-generated training library in MarvinSketch, see the following steps:

1. Select MarvinSketch menu *Tools > Protonation > pKa*.
2. Set the *Use correction library* option to activate the training option (see figure below).
3. If you have created multiple training sets, choose the most accurate one from the dropdown list below the checkbox.

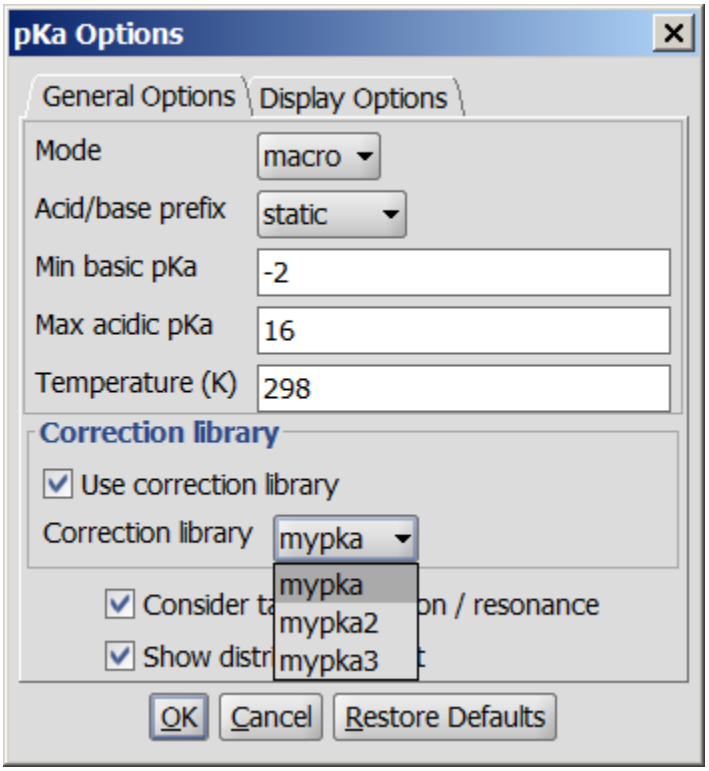
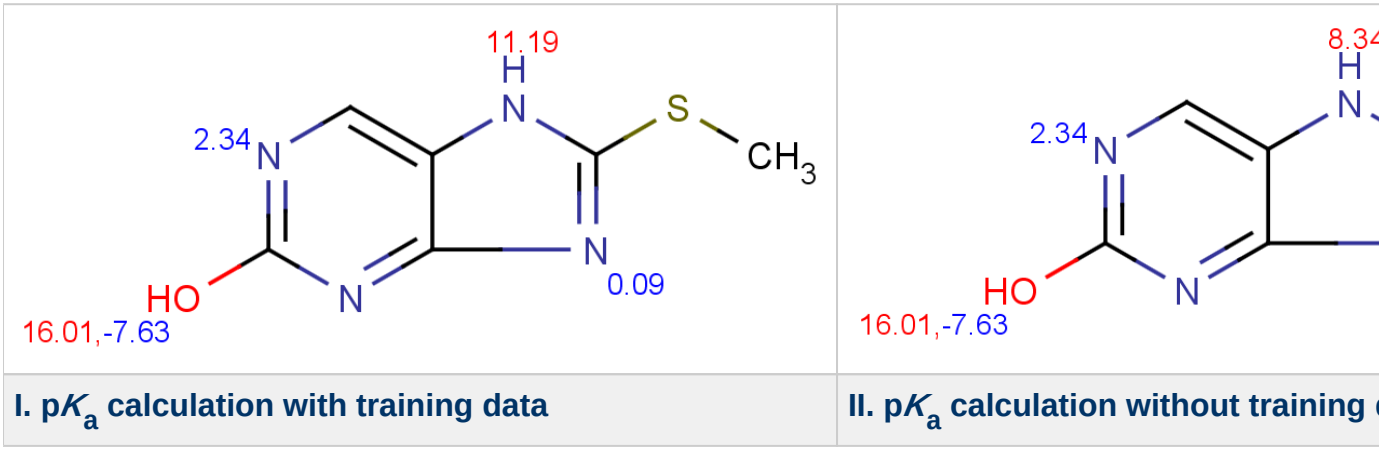


Fig. 2 Using the generated training library in MarvinSketch

The following figure shows the results with (I) and without (II) applying the correction library.



I. pK_a calculation with training data | **II. pK_a calculation without training**

Cxcalc

To include your correction library in the pKa calculation use the parameter *--correctionlibrary* or its short form *-L* :

```
cxcalc pKa  --correctionlibrary  [library name] [input file/string]
```

> If you use cxcalc pKa calculation without the correction library, the results will be calculated with the built-in dataset.

Example #1:

```
cxcalc pKa --correctionlibrary mypka "CSC1=NC2=C(N1)C=NC(O)=N2"
```

Result

id	apKa1	apKa2	bpKa1	bpKa2	atoms
1	11.19	16.01	2.34	-2.59	7,11,9,4

Example #2

```
cxcalc pKa "CSC1=NC2=C(N1)C=NC(O)=N2"
```

Result

id	apKa1	apKa2	bpKa1	bpKa2	atoms
1	8.34	16.01	2.34	-2.59	7,11,9,4

Chemical Terms

Chemical Terms are available from Chemical Terms Evaluator or from Instant JChem. Evaluator is designed to evaluate Chemical Terms expressions on molecules. Your correction library can be applied as follows:

```
evaluate -e "pKa('correctionlibrary:[library name]')" "[input file
/string]"
```

Example

```
evaluate -e "pKa('correctionlibrary:mypka')" "CSC1=NC2=C(N1)C=NC(O)=N2"
```

Result

```
;;;-2,59;;;11,19;;2,34;;16,01;
```

Instant JChem

Choose the 'New Chemical Terms Field icon' and type the chemical term into the window, use the *correctionlibrary:[library name]* parameter. Do not forget to adjust the *Name*, the *Type* and the *DB Column Name*.

Example

The following picture demonstrates the usage of pK_a training in the 'New Chemical terms' window. The expression

```
pKa('correctionlibrary:mypKa type:acidic','1')
```

defines that the plugin use the correction library named *mypKa*, and it calculates the strongest acidic p*Ka* of the molecule(s).

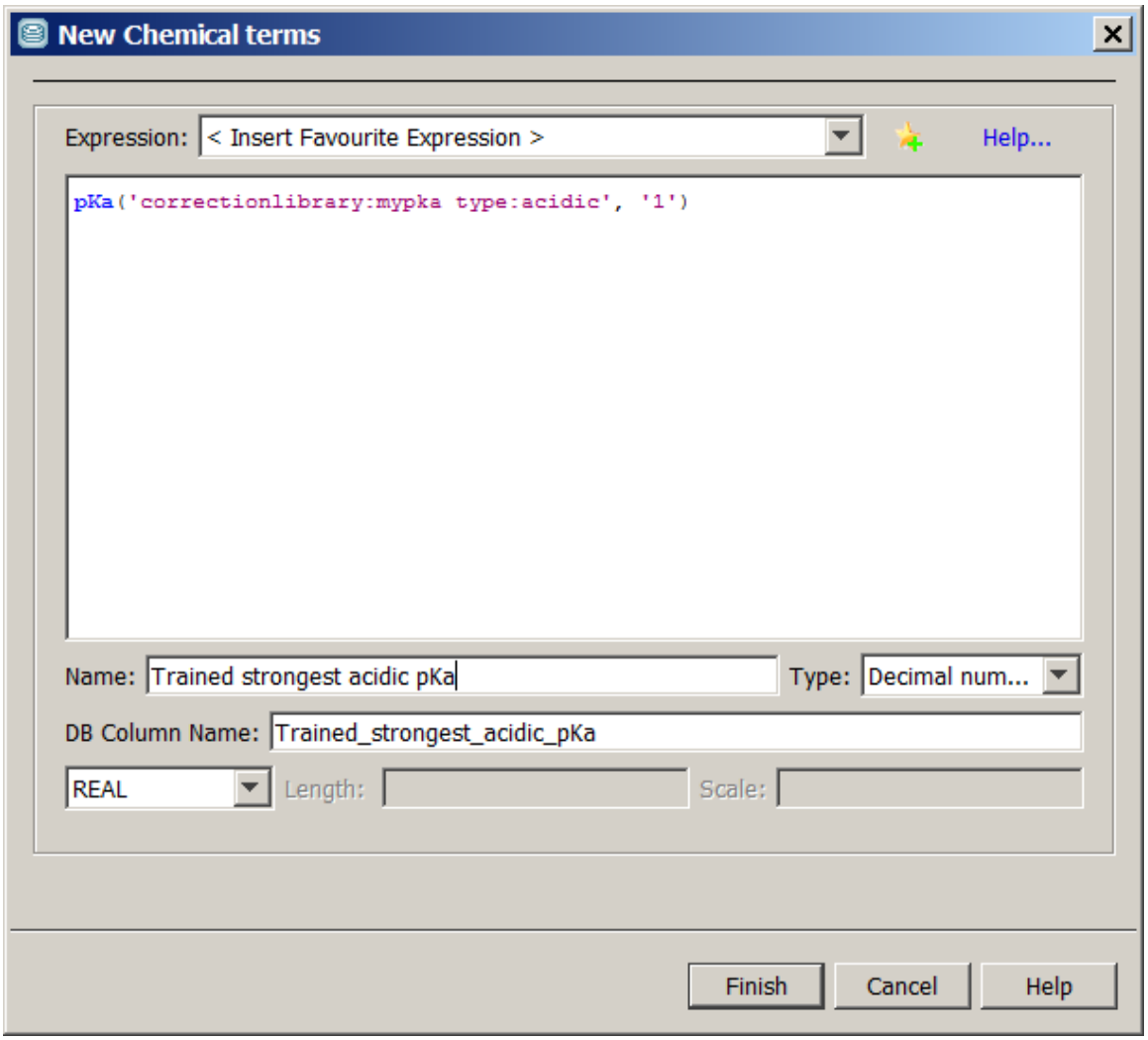


Fig. 3 New Chemical terms window showing the options to be set for pK_a training

The results of this calculation are shown in the figure below, with the untrained (*Strongest acidic pKa* column) and trained (*Trained strongest acidic pKa* column) pK_a values.

Welcome × | Grid view for pKa_trainingset ×

Query | Browse | Code

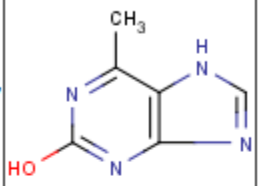
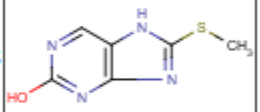
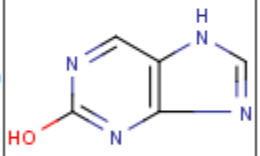
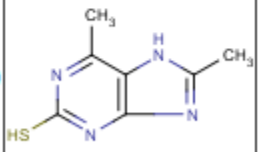
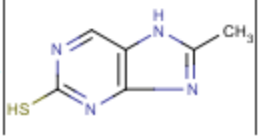
...	CdId	Structure	Mol Weight	Formula	Strongest acidic pKa	Trained strongest
17	17		150,14	C6H6N4O	10,39	12,24
18	18		182,20	C6H6N4OS	8,34	11,19
19	19		136,11	C5H4N4O	9,85	12,09
20	20		180,23	C7H8N4S	9,63	9,66
21	21		166,20	C6H6N4S	9,47	9,55

Fig. 4 JChem table showing the trained and untrained pK_a values