CypScore Pipeline Pilot Components

Table of Contents

1 Introduction

The CypScore Pipeline Pilot Components from CACheResearch LLC provide an implementation of the method devised by Hennemann *et al*. [1] to predict positions in drug-like molecules that are likely sites for metabolism by cytochromes P450 within the Accelrys Pipeline Pilot workflow system.

Based on a hypothetical "P450 super-enzyme", CypScore applies six models for the oxidation reactions listed in Table 1. Each model is a linear equation $y = a_0 + \sum_{i=1}^{4} a_i x_i$ predicting an atom's reactivity y from the values of up to four descriptors x_i calculated by ParaSurf [2]. The descriptors involved in each model and the coefficients a_i are defined in the original publication. The CypScore component assigns a model to each C, N and S atom visible on the molecular surface, and then calculates the value of that model using the appropriate ParaSurf descriptors.

Table 1: CypScore reaction models

Model	Reaction
1	aliphatic hydroxylation, N-dealkylation, O-dealkylation
2	aromatic hydroxylation
3	double-bond epoxidation/oxidation
4a	N-oxidation of amines
4b	N-oxidation of imines
5	S-oxidation

Each of the CypScore models produces reactivity predictions on an individual scale. Hennemann *et al*. [1] described a procedure to establish a common reactivity scale for all the models, providing a CypScore value *S* for each atom in the range from 0 (stable) to 100 (reactive), and implemented this procedure for a proprietary data set. Unfortunately, the CypScore paper does not include sufficient detail to define completely the relationship between the original model values y and the scaled scores *S*, and it is not possible to reproduce the missing information as the data from which it was derived is not available publicly. To overcome this, the CypScore component employs a common reactivity scale that is a close approximation to the one used by Hennemann *et al*. [1], derived from secondary information in the original paper through a process described in section 6. Consequently, the results obtained by the CypScore component will approximate closely, but not match exactly, those in the original publication.

2 Prerequisites

2.1 CEPOS Pipeline Pilot Components

The CEPOS Pipeline Pilot Components, available free of charge from CEPOS InSilico Ltd., are required to provide a Pipeline Pilot interface to ParaSurf [2]. Email support@ceposinsilico.com to request a copy.

2.2 ParaSurf

ParaSurf'10 or a later version is required. ParaSurf is available from CEPOS InSilico Ltd. ParaSurf may be installed either on the Pipeline Pilot server or on the system running the Pipeline Pilot client. It is recommended that ParaSurf is installed on the Pipeline Pilot server.

2.3 Pipeline Pilot

Pipeline Pilot version 7.5 or later is required. Pipeline Pilot is available from Accelrys Software Inc.

2.4 VAMP

ParaSurf requires the results of semi-empirical molecular orbital calculations provided by VAMP [3], which is available from Accelrys Software Inc.

2.5 VAMP Descriptors Component

In the CEPOS Pipeline Pilot Components, VAMP calculations are carried out using the VAMP Descriptors Component from version 4.4 of the Materials Component Collection, which is available from Accelrys Software Inc.

2.6 CEPOS MOPAC6

Customers without access to VAMP may instead use the CEPOS MOPAC6 program. However, use of VAMP in preference to MOPAC is strongly recommended. CEPOS MOPAC6 is available free of charge from CEPOS InSilico Ltd.

3 Obtaining and Installing the Software

The CypScore Pipeline Pilot Components are available free of charge from CACheResearch LLC. Contact info@CACheResearch.com to request a copy. The software is provided as a zip archive *cypscorepp–x.y.zip*, where *x.y* is the version number. To install the CypScore components into Pipeline Pilot follow these steps.

1. First remove any existing installation of the CypScore Pipeline Pilot Components. Run the command **pkgutil –u cache** on the Pipeline Pilot server, then remove the directory *<PP>/apps/cache*, where *<PP>* is the Pipeline Pilot installation directory.

2. On the Pipeline Pilot server, copy *cypscorepp–x.y.zip* to the directory *<PP>/apps*, then unzip it. A directory *<PP>/apps/cache* will be created.

3. Run the command **pkgutil –i cache**. This will install the CypScore components and example protocols in the Pipeline Pilot server. The Components tab in the Pipeline Pilot client will now contain a directory called *CACheResearch* with a subdirectory named *CypScore*. The Protocols tab in the Pipeline Pilot client will contain a directory called *CACheResearch Examples*, also with a *CypScore* subdirectory. Some sample input files for the example

protocols are also installed and can be accessed from the *CypScore Data* shortcut in the Server File Browser.

4 Components

4.1 CypScore

This component implements the CypScore method of Hennemann *et al*. [1]. Each atom is assigned one of six CypScore models and a CypScore value between 0 (stable) and 100 (reactive). These are saved as the atom properties *CypScoreModel* and *CypScore*. Each bond is assigned a CypScore value which is the mean of the non-zero scores of the bond's atoms. This is saved as the bond property *CypScore*. Atom and bond properties can be retrieved either by using the Molecular Toolkit or by using the *Atom Properties* and *Bond Properties* components.

Input molecules for this component must contain a specific set of properties, which are calculated by the Parasurf (on server) and ParaSurf (on client) components. The required properties must be generated with the combination of options used to derive the CypScore models [1], which are:

```
surf=cube estat=multi contour=isoden iso=0.0003
```

These values for estat, contour and iso are the default values if surf=cube is specified, so it is sufficient to run ParaSurf with surf=cube. Note that these options are *not* the default values for the ParaSurf (on server) and ParaSurf (on client) components.

In order for the CypScore model assignment to function correctly, the input to the ParaSurf (on client/server) components must originate from the CEPOS VAMP component. In addition, the *Calculate Bonds* parameter in the CEPOS VAMP component must be set to *False*, as the bond types calculated by VAMP are incompatible with the CypScore component.

4.2 CypScore Report

The CypScore Report component transforms the output from the CypScore component into a table with columns containing the molecule **Name**, **CypScore Structure**, **Atom CypScores** and **Bond CypScores**. Each cell in the **Atom CypScore** column has a row for each heavy atom in the molecule containing three items: the atom number, the model type assigned to the atom (1, 2, 3, 4a, 4b or 5; with 0 indicating that no model was assigned), and the CypScore value for the atom in the range from 0 (stable) to 100 (reactive). Each cell in the **Bond CypScore** column has a row for each pair of bonded heavy atoms containing the atom numbers and the mean scores of the bonded atoms. The **CypScore Structure** shows the 2D structure of the molecule with atoms labelled by their CypScore values. An example is shown in Figure 1. Atoms with scores of 38 and above are considered reactive and coloured red, atoms with scores below 22 are considered stable and coloured black, atoms with intermediate scores are coloured blue. The size of the images is controlled by the *Image Size* parameter.

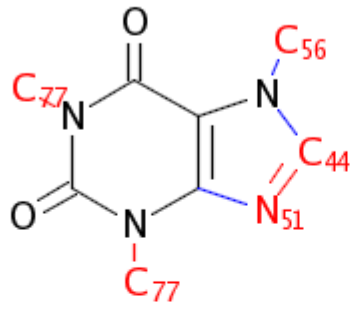
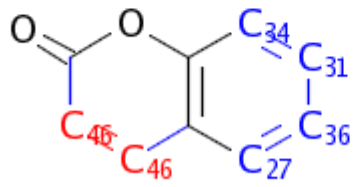
Name	CypScore Structure	Atom CypScore	Bond CypScore
Caffeine		<table><tr><th>atom</th><th>model</th><th>score</th></tr><tr><td>0</td><td>2</td><td>10.00</td></tr><tr><td>1</td><td>2</td><td>0.00</td></tr><tr><td>2</td><td>0</td><td>0.00</td></tr><tr><td>3</td><td>4a</td><td>0.00</td></tr><tr><td>4</td><td>4a</td><td>0.00</td></tr><tr><td>5</td><td>4b</td><td>51.36</td></tr><tr><td>6</td><td>4a</td><td>0.00</td></tr><tr><td>7</td><td>0</td><td>0.00</td></tr><tr><td>8</td><td>2</td><td>44.06</td></tr><tr><td>9</td><td>1</td><td>56.28</td></tr><tr><td>10</td><td>0</td><td>0.00</td></tr><tr><td>11</td><td>1</td><td>77.46</td></tr><tr><td>12</td><td>1</td><td>77.01</td></tr><tr><td>13</td><td>0</td><td>0.00</td></tr></table>	<table><tr><th>atom1</th><th>atom2</th><th>score</th></tr><tr><td>1</td><td>2</td><td>5.00</td></tr><tr><td>1</td><td>3</td><td>5.00</td></tr><tr><td>1</td><td>4</td><td>5.00</td></tr><tr><td>2</td><td>5</td><td>0.00</td></tr><tr><td>2</td><td>6</td><td>25.68</td></tr><tr><td>3</td><td>7</td><td>0.00</td></tr><tr><td>3</td><td>8</td><td>0.00</td></tr><tr><td>4</td><td>9</td><td>22.03</td></tr><tr><td>4</td><td>10</td><td>28.14</td></tr><tr><td>5</td><td>11</td><td>0.00</td></tr><tr><td>5</td><td>12</td><td>38.73</td></tr><tr><td>6</td><td>9</td><td>47.71</td></tr><tr><td>7</td><td>13</td><td>38.51</td></tr><tr><td>7</td><td>11</td><td>0.00</td></tr><tr><td>11</td><td>14</td><td>0.00</td></tr></table>
Coumarin		<table><tr><th>atom</th><th>model</th><th>score</th></tr><tr><td>0</td><td>2</td><td>0.00</td></tr><tr><td>1</td><td>2</td><td>0.00</td></tr><tr><td>2</td><td>0</td><td>0.00</td></tr><tr><td>3</td><td>2</td><td>33.75</td></tr><tr><td>4</td><td>3</td><td>46.40</td></tr><tr><td>5</td><td>2</td><td>27.47</td></tr><tr><td>6</td><td>0</td><td>0.00</td></tr><tr><td>7</td><td>2</td><td>31.30</td></tr><tr><td>8</td><td>3</td><td>46.40</td></tr><tr><td>9</td><td>2</td><td>35.67</td></tr><tr><td>10</td><td>0</td><td>0.00</td></tr></table>	<table><tr><th>atom1</th><th>atom2</th><th>score</th></tr><tr><td>1</td><td>2</td><td>0.00</td></tr><tr><td>1</td><td>3</td><td>0.00</td></tr><tr><td>1</td><td>4</td><td>16.87</td></tr><tr><td>2</td><td>5</td><td>23.20</td></tr><tr><td>2</td><td>6</td><td>13.73</td></tr><tr><td>3</td><td>7</td><td>0.00</td></tr><tr><td>4</td><td>8</td><td>32.52</td></tr><tr><td>5</td><td>9</td><td>46.40</td></tr><tr><td>6</td><td>10</td><td>31.57</td></tr><tr><td>7</td><td>11</td><td>0.00</td></tr><tr><td>7</td><td>9</td><td>23.20</td></tr><tr><td>8</td><td>10</td><td>33.49</td></tr></table>

Figure 1. Sample output from the CypScore Report component displayed in the HTML Report Viewer.

5 Example Protocols

The example protocols provided with the CypScore Pipeline Pilot Components run CypScore from various starting points, illustrate the use of the reporting components and show how to manipulate the output from the CypScore components. When constructing protocols with the CypScore components, bear in mind that the CypScore models were trained on 3D structures generated by CORINA and optimized with VAMP. For best results, the same procedure should be followed when applying the models.

5.1 CypScore from SDF

The CypScore from SDF protocol processes SD files containing either 2D or 3D coordinates. If 2D coordinates are provided, the molecules are prepared by generating 3D coordinates and minimizing the structures. The CEPOS VAMP, ParaSurf and CypScore components are then run in turn, and the results processed by the CypScore Report component. The intermediate ParaSurf output files (*_p.out* and *_p.sdf*) and a CypScore SD file (*_cyp.sdf*), containing the CypScore atom and bond scores, are saved in the Job Directory.

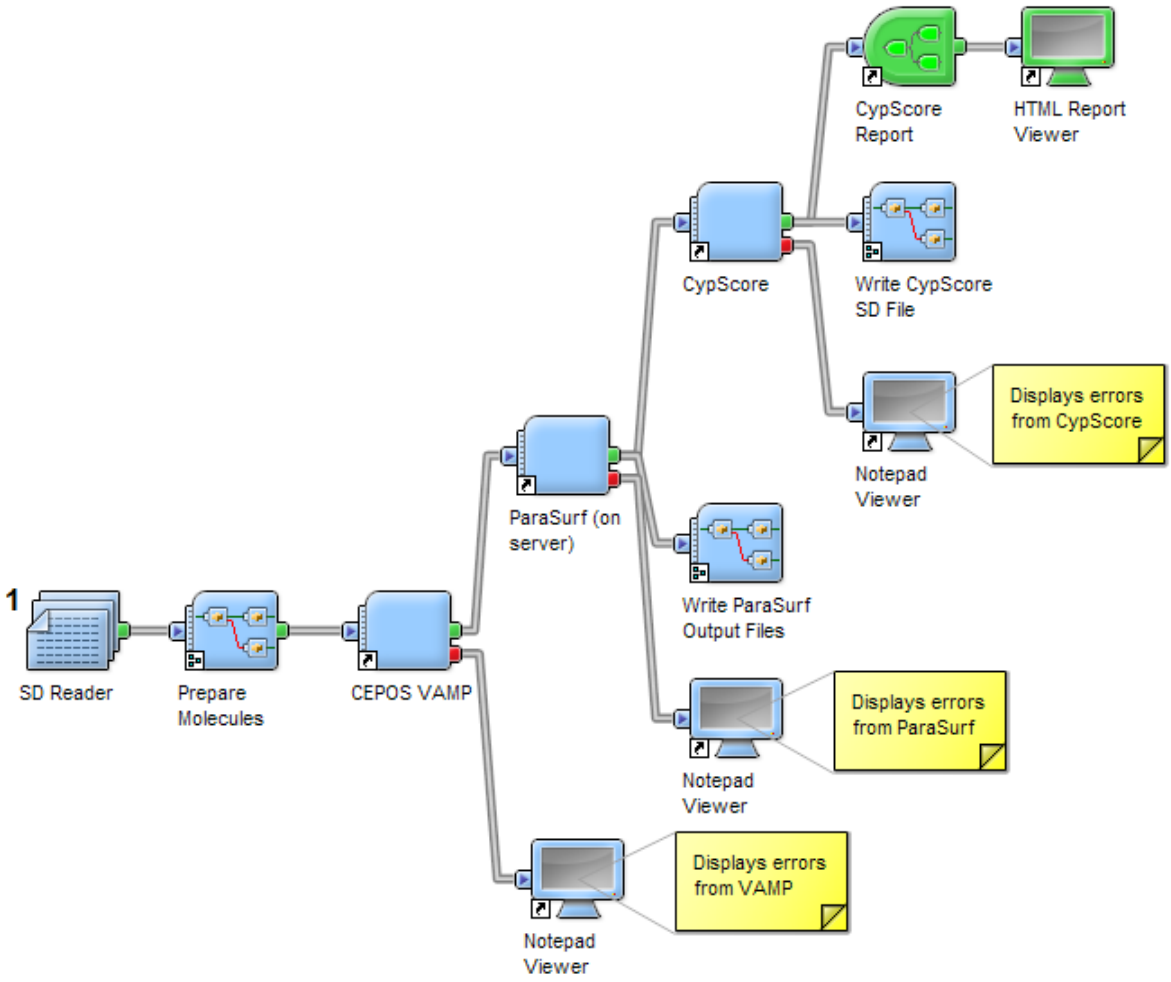


Figure 2. The CypScore from SDF protocol.

5.2 CypScore from SMILES

The CypScore from SMILES protocol performs a CypScore calculation starting from a SMILES file. The molecules are prepared by generating 3D coordinates and minimizing the structures. The CEPOS VAMP, ParaSurf and CypScore components are then run in turn. The results are processed by the CypScore Report component and displayed in an HTML Report Viewer. The intermediate ParaSurf output files (*_p.out* and *_p.sdf*) and a CypScore SD file (*_cyp.sdf*), containing the CypScore atom and bond scores, are saved in the Job Directory.

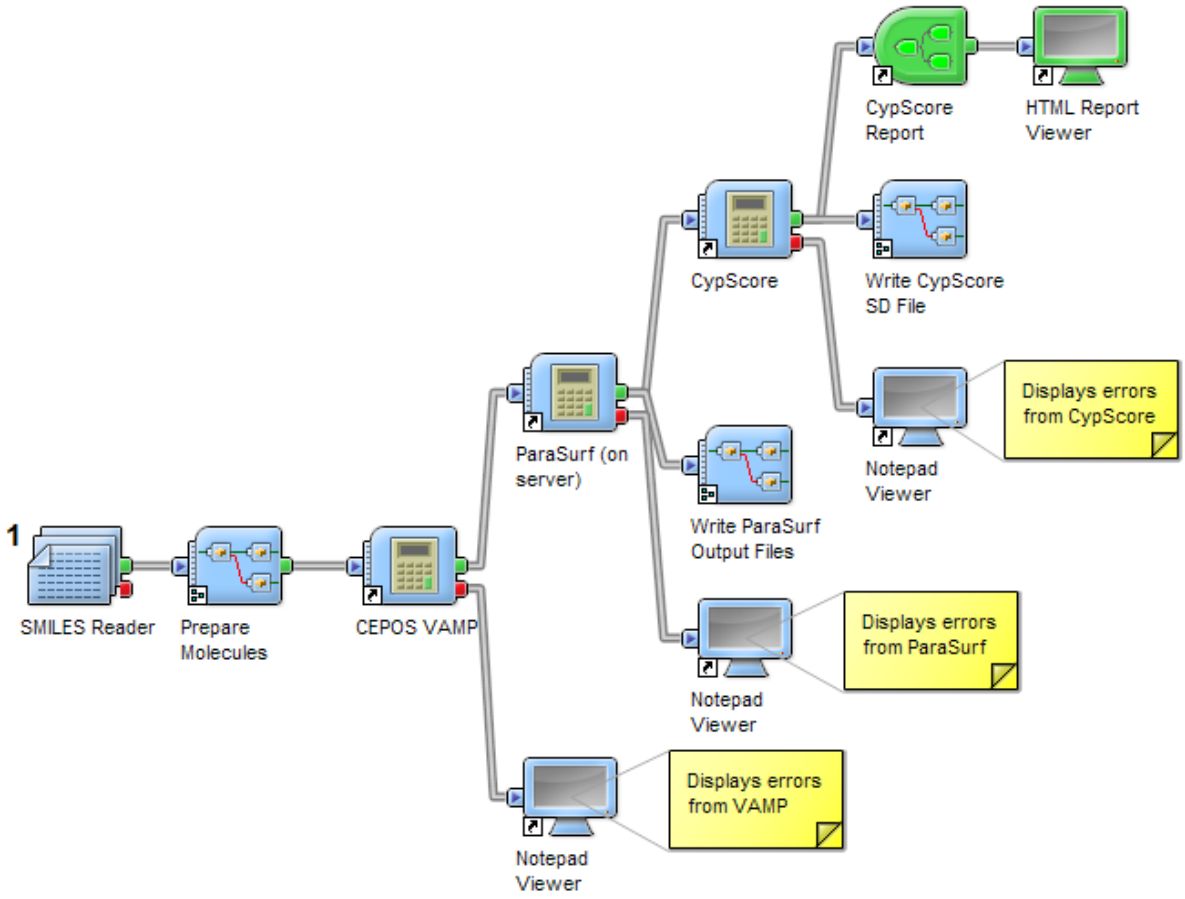


Figure 3. The CypScore from SMILES protocol.

6 CypScore Scaling

To express the reactivity predictions of the CypScore models on a common scale, Hennemann *et al.* [1] employ an empirical procedure based on the enrichment curves obtained for the dataset that was used to train the models. The cross-over value where the false metabolic rate equals the false stable rate is taken as a common reference point for each model. The cross-over points for the individual models are aligned and the range of values for each model is scaled and truncated to lie between 0 (representing stable atoms) and 100 (representing highly reactive atoms).

Specifically, the following sequence of transformations is applied. For a particular model, let y denote the raw score given by the corresponding equation in the paper. First, the cross-over point C for the false metabolic and false stable rates is translated to the origin, defining $y_1 = y - C$. Next, the range $[\max y_1 , \min y_1]$ is scaled linearly to fit the interval [-10, 10]. Thus define y_2 by

$$y_2 = \begin{cases} -10y_1/\max y_1 , & \max y - C \geq C - \min y \\ 10y_1/\max y_1 , & \max y - C < C - \min y \end{cases}$$

Finally, the range [-7, 3] for y_2 is mapped linearly to [0, 100] and values outside this range are mapped to the endpoints. Hence the scaled score S is defined by

$$S = \begin{cases} 0, & y_2 \geq 3 \\ -10y_2, & -7 < y_2 < 3 \\ 100, & y_2 \leq -7 \end{cases}$$

For each model, if the transformation from y to y_2 is written as $y_2 = a(y - b)$ then the map from y to S can be written

$$S = \begin{cases} 0, & y \leq y_{min} \\ Ay + B, & y_{min} < y < y_{max} \\ 100, & y \geq y_{max} \end{cases} \qquad (*)$$

where $A = -10a$, $B = 10ab + 30$, $y_{min} = b + (3/a)$ and $y_{max} = b - (7/a)$.

Hennemann *et al.* [1] followed this procedure using the false positive and false negative rates for a proprietary dataset and the values of *a* and *b* (or their equivalents) obtained for this dataset are not revealed. Hence the precise scaling used in the paper is not known.

However, an approximation to the common reactivity scale can be recovered from a careful examination of Figure 3 in Hennemann *et al.* [1]. This figure contains four plots, each showing the scores for one of the four validation datasets used in the paper: three public domain datasets provided as supporting information (set0, set1 and set2), and one proprietary dataset (set3). Each column in these plots depicts the scores for the atoms in a single molecule as coloured squares, with reactive atoms in blue and stable atoms in red. The unscaled model scores for the public domain datasets can be found by following the protocols described in the paper. By comparing the plots for the public

domain datasets in Figure 3 with the unscaled model scores, it is then possible to detect, approximately, the scaling parameters in equation (*).

The difficulty in doing this is that while each column in Figure 3 contains the set of scores for a single molecule, there is no information to show which model was used for a particular data point; the scores from the six models are interleaved down each column in the figure. The task is therefore to match the data points in the figure with specific atoms. Provided sufficient matches can be made for atoms assigned to a particular model, a simple linear regression of the scaled against the unscaled scores will then reveal the scaling transformation for that model.

To accomplish this, numerical scores were extracted from Figure 3 for set0 and set2 by matching the data points in the figure, by hand, to the nearest point on a scale with intervals of length 0.5. (Set1 was not used as its plot is so dense that individual data points cannot be distinguished reliably; in fact, it is difficult even to detect which columns many data points lie in.)

An iterative strategy was then followed to match data points to atoms. First, reactive atoms in set0 and set2 (those with CYP entries in the SD files supplied as supporting information) were matched, as far as possible, to blue points in the figure. Next, the higher scoring atoms were examined for cases where there was a clear correspondence between atoms and data points. Some molecules, for instance, have models of type 1 or type 2 only. After a first pass through set0 and set2, very rough relationships between the raw and scaled scores emerged for models 1 and 2, which form the majority of the cases. A second pass through set0 and set2 was then made to reassign the worst outliers in the first pass model 1 and model 2 results. This in turn allowed more atoms to be assigned to data points and the process was repeated until a sufficient number of matches were established for atoms assigned to each model.

This laborious process is prone to various types of possible error: differences in atom typing and model assignment, the mismatch in the numbers of atoms with non-zero scores in the validation sets and the number of visible points in Figure 3 (some points are evidently plotted on top of one another), and inaccuracies in assigning numerical scores to the data points in the figure. Nevertheless, convincing solutions were found for all the models, and the resulting scaling parameters, using the notation of equation (*), are shown in Table 2, along with the number of atoms assigned for each model (N) and the R^2 value for the linear fit of S against y. Here y_{min} and y_{max} are given by $y_{min} = -B/A$ and $y_{max} = (100 - B)/A$.

Table 2: CypScore scaling parameters

Model	A	B	y_{min}	y_{max}	N	R^2
1	236	-19.6	0.083	0.51	279	0.98
2	497	-12.7	0.026	0.23	194	0.92
3	268	0.19	-0.001	0.37	14	0.99
4a	840	-26.3	0.031	0.15	24	0.90
4b	778	-13.6	0.017	0.15	11	0.96

5	107	-7.65	0.071	1.01	8	0.93

7 Support

7.1 Contact

Questions regarding the CypScore Pipeline Pilot Components should be addressed to:

info@CACheResearch.com

7.2 CAChe Research LLC.

Americas
CAChe Research LLC
Oregon, USA
Email: Info@CACheResearch.com
Tel: +1 503 830 2772
Fax: +1 206 203 4405

Europe
CAChe Research LLC
Somerset, UK
Email: Info@CACheResearch.com
Tel: +44 2081 444080

Web: www.CACheResearch.com

8 References

[1] M. Hennemann, A. Friedl, M. Lobell, J. Keldenich, A. Hillisch, T. Clark and A. H. Goller. *CypScore: quantitative prediction of reactivity toward cytochromes P450 based on semiempirical molecular orbital theory*. ChemMedChem, 2009, 4, 657-669.

[2] A. H. C. Horn, J.-H. Lin and T. Clark. *A multipole electrostatic model for NDDO-based semiempirical molecular orbital methods. Theor. Chem. Accts.*, 2005, *113*, 159-168. Erratum: *Theor. Chem. Accts*., 2007, *117*, 461-465.

[3] T. Clark, A. Alex, B. Beck, F. Burkhardt, J. Chandrasekhar, P. Gedeck, A.H.C. Horn, M. Hutter, B. Martin, G. Rauhut, W. Sauer, T. Schindler and T. Steinke . *VAMP 11.0.* Erlangen 2008. Available from Accelrys Inc., San Diego, USA. (www.accelrys.com/products/materials-studio/modules/VAMP.html)