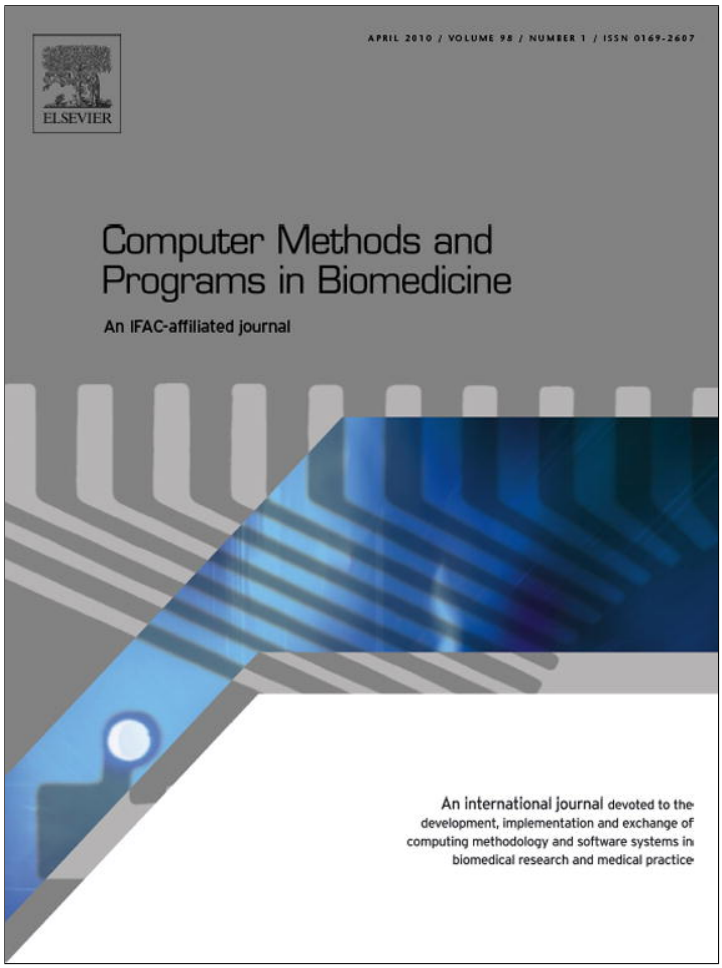
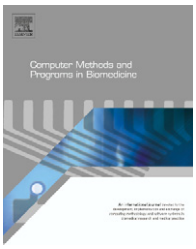


COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 98 (2010) 45–54

journal homepage: www.intl.elsevierhealth.com/journals/cmpb



A ligand predication tool based on modeling and reasoning with imprecise probabilistic knowledge☆

Weiru Liu[a,*], *Anbu Yue*[a], *David J. Timson*[b]

[a] *School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT7 1NN, UK*
[b] *School of Biological Science, Queen's University Belfast, Belfast BT9 7BL, UK*

ARTICLE INFO

Article history:
Received 8 April 2009
Received in revised form
18 August 2009
Accepted 24 August 2009

Keywords:
Imprecise probabilistic knowledge
Prediction tool
Substrate structure
Enzymes

ABSTRACT

Ligand prediction has been driven by a fundamental desire to understand more about how biomolecules recognize their ligands and by the commercial imperative to develop new drugs. Most of the current available software systems are very complex and time-consuming to use. Therefore, developing simple and efficient tools to perform initial screening of interesting compounds is an appealing idea.

In this paper, we introduce our tool for very rapid screening for likely ligands (either substrates or inhibitors) based on reasoning with imprecise probabilistic knowledge elicited from past experiments. Probabilistic knowledge is input to the system via a user-friendly interface showing a base compound structure. A prediction of whether a particular compound is a substrate is queried against the acquired probabilistic knowledge base and a probability is returned as an indication of the prediction.

This tool will be particularly useful in situations where a number of similar compounds have been screened experimentally, but information is not available for all possible members of that group of compounds. We use two case studies to demonstrate how to use the tool.

1. Introduction

Considerable investment has been made into the in silico prediction of substrates, and especially, inhibitors of enzymes. This investment has been driven by a fundamental desire to understand more about how biomolecules recognize their ligands and by the commercial imperative to develop new drugs. Almost all pharmaceutical companies include an element of target-based approaches in their drug discovery programmes. In this paradigm a key biomolecule (usually an enzyme or receptor), the "target", is identified and characterized. A major effort is then put into discovering small molecules which will modify the activity of the target in a therapeutically useful way. For example, if the target is a vital and unique enzyme in an infectious microorganism, the aim may be to discover molecules which act as high affinity inhibitors. A variety of approaches are used to identify suitable molecules including high throughput screening of compound libraries against the target and computational screening of molecules. These processes are both time-consuming and expensive. Typical estimates suggest that bringing a novel drug successfully to market costs approximately $1 billion and takes 10–20 years. Clearly there is a need to introduce new methods to increase the speed at which potential drug molecules can be discovered. In addition, the explosion of sequence data in recent years (primarily resulting from genome sequencing projects), has identified a large number of enzymes (etc.) whose func-

☆ *Note:* A preliminary version reporting the theoretical development (not the tool) of this work was presented at the 5th International Symposium on Imprecise Probability: Theories and Applications, University of Durham, UK, 14–18 July 2009.

* *Corresponding author*. Tel.: +44 289097 4986.
E-mail addresses: w.liu@qub.ac.uk (W. Liu), a.yue@qub.ac.uk (A. Yue), d.timson@qub.ac.uk (D.J. Timson).

0169-2607/$ – see front matter © 2009 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2009.08.009

tions can be guessed at (by application of sequence similarity searches) but for which experimental determination of their substrate specificity and range will be required. Again this is likely to be a time-consuming and expensive process.

Modern programs are increasingly sophisticated and incorporate quantum mechanical parameters, attempt to estimate entropic contributions of the solvent and allow for the inherent flexibility of biological macromolecules. Nevertheless, despite this complexity, their predictions are not perfect. Furthermore, with increasing sophistication comes increased computational times. Thus investigators will generally only carry out detailed computational studies on molecules which are either known to be good ligands (by experiments) or which have been predicted by less complex programmes as being highly likely to be good ligands.

Probabilistic logic programming is a framework to represent and reason with imprecise (conditional) probabilistic knowledge. An agent's knowledge is represented by a *probabilistic logic program* (PLP) which is a set of (conditional) logical formulae with probability intervals. The impreciseness of the agent's knowledge is explicitly represented by assigning a probability interval (or a single probability) to every logical formula indicating that the probability of the formula shall be in the given interval. For example, $(fly(X)|bird(X))[0.8, 1.0]$ is a probabilistic formula which says that under the condition that if object X is a bird then the probability of X can fly is within [0.8, 1] statistically.

Probabilistic logic programming has been used to represent and reason with probabilistic knowledge in many real-world applications, e.g. [1–3]. In general, given a PLP, there is a set of probability distributions satisfying the probabilistic knowledge in the PLP. One approach to determining a single unique probability distribution among all these possible distributions is to apply the maximum entropy principle.

The maximum entropy principle is a widely accepted method for probabilistic reasoning. It can be applied to reason with compatible probabilities implied by an PLPs [4]. Based on this principle, a single probability distribution is selected and the selected distribution has the maximum entropy value among all possible probability distributions. This probability distribution is assumed to best fit the imprecise probabilistic knowledge in the PLP.

In this paper, we present our investigation about how to use PLPs to represent and reason with imprecise probabilistic knowledge obtained from experiments, especially on substrates prediction in biomedical sciences. We present our implementation of a probabilistic prediction system which takes PLPs as input knowledge bases and produces probabilistic results for queries (against a chosen PLP) as output.

To facilitate bioscientists using this system, we implemented a prediction tool for very rapid screening for likely ligands (either substrates or inhibitors) based on reasoning with imprecise probabilistic knowledge elicited from past experiments. This tool has a user-friendly interface which allows a user to select a base-compound structure to start with. For example, if a user selects *sugar*, then the system will show the base structure associated with *sugar*. This way, a user can immediately move on adding any additional compound information as well as any probabilistic information to this structure, saving both time and effort to build a knowledge base. A generated PLP from a user's input is then displayed on the screen for inspection. When satisfied, a user can then pose any queries to the knowledge base to predict other possible compounds. The saved knowledge base (PLP) can be repeatedly used by any users who can access this knowledge base.

We conducted two sets of experiments using the tool, one is on the human enzyme galactokinase (EC 2.7.1.6), which uses galactose as a substrate, and the other is on substrate prediction for human NAD(P)H quinone oxidoreductase (EC 1.6.5.2, NQO1). The experimental results demonstrate that using imprecise probabilistic knowledge as a first step for screening potential substrates can be very useful and significant in many similar applications, since this initial prediction could allow bioscientists to selectively experiments on more hopeful candidates, saving both time and money in the whole process.

This paper is organized as following. In Section 2, we briefly review the modelling method used in this paper for representing probabilistic knowledge. In Section 3, we first introduce the architecture of our tool and the main system functionalities. We then give some detailed descriptions of the graphical user interface on how to visually model substrate knowledge in our system, the input, output and query format, as well as how we use XML files to store information about base structures. In Section 4, we illustrate how to use our prediction tool with two case studies. Finally, we conclude this paper after comparing our tool with some related systems in Section 5.

2. Computational methodology and theory: probabilistic logic programming

We briefly review conditional PLPs here [4,5].

Let Φ be a finite set of *predicate symbols* and c onstant symbols, and $\mathcal{V}$ be a set of *variables*. An *event* or *logic formula* can be defined from $\Phi \cup \mathcal{V}$ using none or any connectives $\neg, \wedge, \vee$ as usually done in first-order logics. We use Greek letters ϕ, ψ, φ for events. For instance, let *Peter* be a person's name, then *man*(*Peter*) is a logical formula saying that *Peter is a man* or let X be a variable, then *man*(X) states that predicate *man* is applied to variable X. Equally, given a constant *talose* and a predicate *aldohexose*, *aldohexose*(*talose*) is a logic formula stating that talose is a (kind of) aldohexose. The Herbrand semantics commonly used in first-order logics can also be canonically defined. An assignment σ maps each variable (in one or more statements) to a constant symbol, such that *man*(*Peter*) can be considered as the result of assigning *Peter* to X.

We use I to stand for a *possible world*, and use $I \models_\sigma \phi$ to state that I is a model of formula ϕ under assignment σ. A *conditional event* is of the form $\psi|\varphi$ with events ψ and ϕ. A *p* robabilistic formula is of the form $(\psi|\varphi)[l, u]$ which means that the probability of conditional formula $(\psi|\varphi)$ is within $[l, u]$. In the following, we call $[l, u]$ the probability bound for probabilistic conditional event $\psi|\varphi[l, u]$. For instance $(fly(X)|bird(X))[a, b]$ states that the probability that a bird can fly falls in the interval $[a, b]$. A *conditional probabilistic logic program* (PLP) P is a set of probabilistic formulae.

A *probabilistic interpretation* Pr is a probability distribution on the set of all possible worlds, which is denoted as $\mathcal{I}_\Phi$. The *probability* of an event φ in Pr is defined as $Pr(\varphi) = \sum_{I \in \mathcal{I}_\Phi, I \models_\sigma \varphi} Pr(I)$.

If $Pr(\phi) > 0$, then $Pr(\psi|\phi)$ exists and it is defined as $Pr(\psi|\phi) = Pr(\psi \bigwedge \phi)/Pr(\phi)$. A probabilistic formula $(\psi|\varphi)[l, u]$ is satisfied by a probabilistic interpretation Pr under assignment σ, denoted as $Pr \models_\sigma (\psi|\varphi)[l, u]$, iff $Pr(\varphi) = 0$ or $Pr(\psi|\varphi) \in [l, u]$. A probabilistic interpretation Pr is a *p* robabilistic model of a formula $(\psi|\varphi)[l, u]$ iff $(\psi|\varphi)[l, u]$ is satisfied by Pr under all assignments, and this is denoted as $Pr \models (\psi|\varphi)[l, u]$. A probabilistic interpretation is a *probabilistic model* of a PLP P, denoted as $Pr \models P$, iff Pr is a probabilistic model of all $\mu \in P$. A PLP P is *satisfiable* or *c* onsistent iff P has a model. A probabilistic formula $(\psi|\varphi)[l, u]$ is a *consequence* of an PLP P, denoted as $P \models (\psi|\varphi)[l, u]$, iff all probabilistic models of P are also probabilistic models of $(\psi|\varphi)[l, u]$. A probabilistic formula $(\psi|\varphi)[l, u]$ is a *tight consequence* of P, denoted as $P \models_{tight} (\psi|\varphi)[l, u]$, iff $P \models (\psi|\varphi)[l, u]$, $P \not\models (\psi|\varphi)[l', u]$, $P \not\models (\psi|\varphi)[l', u]$ for all $l < l'$ and $u > u'$ $(l', u' \in [0, 1])$. Note that, if $P \models (\phi|\top)[0, 0]$, then it is canonically defined as $P \models_{tight} (\psi|\phi)[1, 0]$, where $[1, 0]$ stands for an empty set.

The principle of maximum entropy is a well known technique for representing probabilistic knowledge. Given a distribution Pr, its entropy quantifies its indeterminateness and it is formally defined as $H(Pr) = -\sum_{I \in \mathcal{I}_\Phi} Pr(I) \log Pr(I)$. Given a PLP P, the *principle of maximum entropy model* (or *me-model*) under σ, denoted by $me_\sigma[P]$, is defined as:

$$H(me_\sigma[P]) = \max H(Pr) = \max_{Pr \models_\sigma P} - \sum_{I \in \mathcal{I}_\Phi} Pr(I) \log Pr(I)$$

$me_\sigma[P]$ is the unique probabilistic interpretation Pr which is a probabilistic model of P under σ and which has the greatest entropy among all the probabilistic models of P under σ.

Let P be a PLP, we say that $(\psi|\varphi)[l, u]$ is a *m* e-consequence of P under σ, denoted as $P \models^{me}_\sigma (\psi|\varphi)[l, u]$, iff P is unsatisfiable, or $me_\sigma[P] \models_\sigma (\psi|\varphi)[l, u]$. We say that $(\psi|\varphi)[l, u]$ is a *tight me-consequence* of P under σ, denoted by $P \models^{me}_{\sigma,tight} (\psi|\varphi)[l, u]$, iff one of the following conditions holds:

- $P \models_\sigma (\varphi|T)[0, 0]$, $l = 1$, $u = 0$,
- $me_\sigma[P](\varphi) > 0$ and $me_\sigma[P](\psi|\varphi) = l = u$.

Example 1. Let PLP P be defined as follows:

$$P = \left\{ \begin{array}{l} (fly(X)|bird(X))[0.98, 1] \\ (bird(X)|penguin(X))[1, 1] \\ (penguin(X)|bird(X))[0.1, 1] \end{array} \right\}$$

Based on this knowledge base, a user can query about the likelihood that *a penguin can fly*, e.g., *?*(*fly*(t)|*penguin*(t)).
The results of using our prediction tool based on this knowledge base is

$P \models_{\sigma,tight} (fly(t)|penguin(t))[0, 1]$,and
$P \models^{me}_{\sigma,tight} (fly(t)|penguin(t))[0.98, 0.98]$.

3. A substrate prediction system

3.1. System architecture

Here we describe our system architecture for rapid substrate prediction. The system is composed of two main parts: a graphical user interface and a prediction engine. The overall architecture of our system is illustrated in Fig. 1.

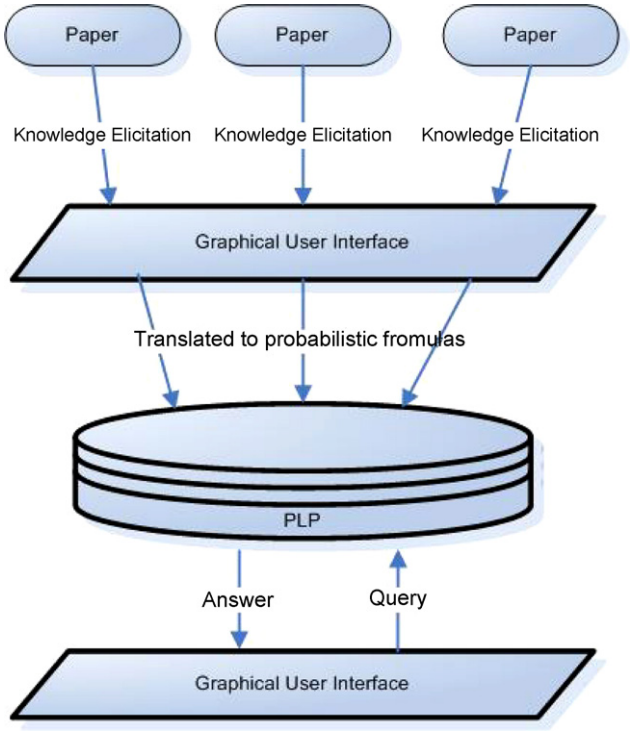


Fig. 1 – The process of substrate predicting with our system.

The prediction engine runs a user specified query against a knowledge base and outputs the prediction result. A knowledge base can be an existing one or a newly created one by a user for the purpose of this query. A newly created knowledge base is saved as part of knowledge base repository and any knowledge base in the knowledge base repository can be revised using an interface component.

The interface consists of several functional components (see Fig. 2). The first is the knowledge base creation component (using the *File* menu) which allows a user to input a new knowledge base where new probabilistic knowledge is elicited (summarized) from experiments in published papers. To facilitate the input of such knowledge, a graphical interface is created allowing a user to visualize the base structure first and then add any additional information. This graphical input, together with any background knowledge that states the compound structure, is automatically translated into a PLP. The reasoning about the PLP for prediction is encapsulated and executed at the back-end, and users do not need to interact with it.

The second component is the knowledge editing component (using the *Edit* menu). A user can navigate all compounds and modify them, and can also insert new compounds with probability intervals. The component structure and the probabilistic knowledge is displayed in the Navigation area. A revised knowledge base is automatically translated into a PLP again to either overwrite the existing one or to create a new one. The translated PLP is displayed just below the Navigation area.

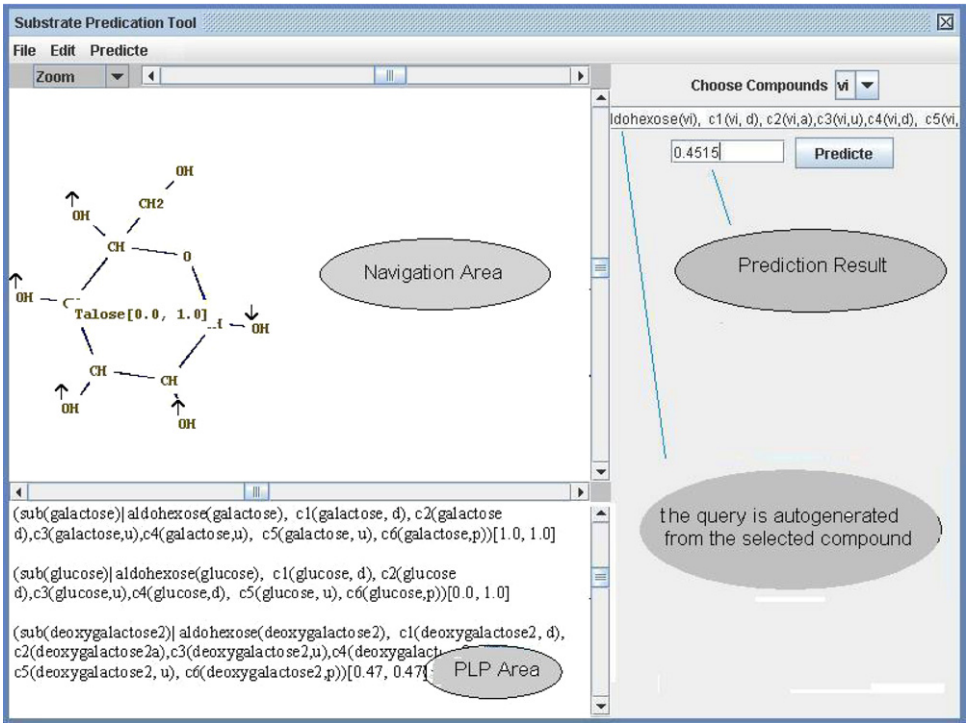


Fig. 2 – The compounds and the corresponding PLP.

The third component is about the input facility of queries. A user is given the option to either construct a new query using the graphical interface by drawing a specific compound or to select an existing query by viewing the collection of compounds stored in the system. In the right-hand column of Fig. 2, the drop-down menu lists all the available queries in the system. When a query is selected, its logic expression is displayed. When a user wants to create a new query, the user draws the compound in the Navigation area and then saves it. Once it is saved, the user can then select it from the drop-down list. By saving previous user specified queries in the system and creating their graphical structures, a user can avoid the task of specifying the same query again, should a user want to re-run the query. The probability of a prediction is then displayed once a user clicks the *Predict* button.

The system is programmed mainly in Java with the integration of some functional components provided in Matlab ™ using EMF (Eclipse Modeling Framework) and Touchgraph ™ libraries. Within the back-end prediction engine, we implemented the efficient algorithms provided in [4,5] for speeding up the prediction. In these algorithms, the problem of reasoning about probabilistic formulae in a PLP is translated into an equivalent problem of solving a (non-linear) optimization subject to linear constraints (using the probability bounds). The optimization problem is then fed into a Matlab optimizer to obtain a set of probability distributions that are compatible with the PLP. A preliminary version of the tool was demonstrated in the European Conference on Artificial Intelligence 2008 Demo session.

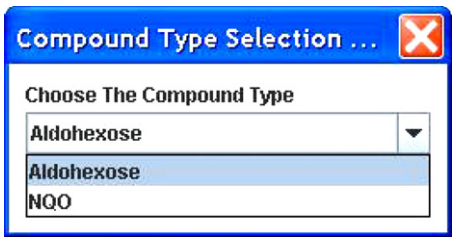


Fig. 3 – Select the compound family by names.

3.2. *Representing substrate structure*

In general, many organic compounds share the same or similar molecular formulas. The organic compounds that have the same or similar molecular formula can be categorized into a class, for instance, the family of aldohexose has 16 stereoisomers. In our tool, the common structure of a family of organic compounds, which we call it the *b* ase structure, is stored as an XML file. When a user wants to create a knowledge base using a base structure stored in the system, the user can select the name of the base structure as shown in Fig. 3, and this file is uploaded to generate the graphical display of the base structure. Then the user has the option to instantiate each base atom (e.g. carbon atom and oxygen atom) and to add any substituent (e.g. –OH group) attached to it.

For instance, each galactose molecule is arranged as a hexagonal ring (e.g., the α-D-galactose molecule in Fig. 4). There are six carbon atoms in a galactose molecule and one oxygen atom. These six carbon atoms are numbered from 1 to 6 with the right-most carbon atom numbered 1, and then the remaining carbons are numbered clockwise round the ring. The oxygen atom is not numbered. The other atoms can be regarded as coming off these carbon atoms. The first four of the carbon atoms each has an OH molecule attached to it, and

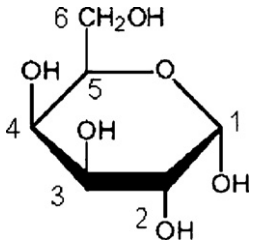


Fig. 4 – The α-D-galactose molecule.

the fifth one has the sixth carbon atom attached to it from outside the ring, forming a CH_2OH group. The OH group can either be "up" or "down" (i.e., they are chiral). The combination of ups and downs gives a specific form of the molecule (in effect, each form of the molecule is a different compound), and the actual combination can significantly affect the biochemical behavior of the molecule. Therefore, for the OH groups attached to these atoms, we need to know if they are *up, down* or *absent*. The sixth carbon is not chiral, and so the OH is neither up nor down. Hence, the OH for the sixth carbon is marked as either *present* or *a* bsent.

In Fig. 5, we present the XML file for aldohexose (α-D-galactose molecule is a specific kind of aldohexose). From this XML file, we can see that, an aldohexose has a cyclic ring with five carbon atoms and one oxygen atom, and an extra carbon atom connected to the carbon in the position 'C6'. The group attached to the carbon atoms is not mandatory, i.e. can be absent.

```
<root>
<Group id = "group", mandatory = "false">
   <Option name = "OH"> OH </Option>
</Group>
<OrganicCompound name = "Aldohexose">
    <Cycle  clockwise = "true">
       <BaseNode  id = "C1" >
           <Option>   C    </Option>
           <Chiral> true </Chiral>
           <Group  ref = "group"/>
       </BaseNode>
       <BaseNode  id = "O" >
           <Option>   O    </Option>
           <Chiral> true </Chiral>
           <Group  ref = "group"/>
       </BaseNode>
       <BaseNode  id = "C2" >
           <Option>  C    </Option>
           <Chiral> true </Chiral>
           <Group  ref = "group"/>
       </BaseNode>
       <BaseNode  id = "C3" >
           <Option> C    </Option>
           <Chiral> true </Chiral>
           <Group  ref = "group"/>
       </BaseNode>
       <BaseNode  id = "C4" >
           <Option> C    </Option>
           <Chiral> true </Chiral>
           <Group  ref = "group"/>
       </BaseNode>
       <BaseNode  id = "C5" >
           <Option> C    </Option>
           <Chiral> true </Chiral>
           <Group  ref = "group"/>
       </BaseNode>
   </Cycle>
   <NonCycle>
       <BaseNode  id = "C6">
          <Option>    C  </Option>
          <Chiral> false </Chiral>
          <Group  ref = "group"/>
          <ConnectTo>
              <BaseNode ref = "C5"/>
          </ConnectTo>
       </BaseNode>
  </NonCycle>
</OrganicCompound>
</root>
```

Fig. 5 – The XML file for sugar family aldohexose.

This XML file indicates that the base atoms in the ring are arranged clockwise and our graphical user interface will display these atoms in the sequence as stated in the file. The base atoms indexed 'C1' to 'C5' are stated as chiral centers (< `Chiral` > true < `/Chiral` >), which means the group attached to them can be either up or down. This piece of knowledge is automatically encoded into probabilistic formulae:

$$(c1(X,p)|aldohexose(X) \bigwedge c1(X,u))[1,1]$$
$$\vdots$$
$$(c5(X,p)|aldohexose(X) \bigwedge c5(X,u))[1,1]$$

The XML element (`mandatory` = "`false`") shows that the group OH that can be attached to 'C1', ..., and 'C6' are non-mandatory, e.g., the group OH can be absent in these positions. Obviously, any group can not be present and absent simultaneously. So this background knowledge is also automatically encoded into probabilistic formulae:

$$(c1(X,p) \bigwedge c1(X,a)|aldohexose(X))[0,0]$$
$$\vdots$$
$$(c6(X,p) \bigwedge c6(X,a)|aldohexose(X))[0,0]$$

In addition, any group that is stated *up* or *down* implies that it is present:

$$(c1(X,p)|aldohexose(X) \bigwedge c1(X,d))[1,1]$$
$$\vdots$$
$$(c5(X,p)|aldohexose(X) \bigwedge c5(X,d))[1,1]$$
$$(c1(X,u) \bigwedge c1(X,d)|aldohexose(X))[0,0]$$
$$\vdots$$
$$(c5(X,u) \bigwedge c5(X,d)|aldohexose(X))[0,0]$$

The predicate $aldohexose(X)$ is needed in the above probabilistic formulae since statements $c1(X,p)$, $c5(X,p)$, etc. are legal only when compound X is an isomer of aldohexose. For instance, $c6(X,u)$ is an illegal logic formula since it is stated in the XML file that the carbon atom indexed as 'C6' is not chiral (so *up* is not applicable to *c6* here).

Once a base structure name is selected, information stored in the corresponding XML file is translated into a graphical structure and this structure is displayed in the Navigation area. Chemical bonds are generated for each pair of base atoms that in the adjacent positions in the same cycle or is stated by the tag `ConnectedTo`. Chemical bonds are also generated for each base atom and the group attached to it. Each group is associated with a pop-up menu, in which a set of options is given showing how that group should be seen within this specific compound.

For instance, the talose in the aldohexose group is displayed in Fig. 6(a). For each organic compound that has chiral centers, we use either the up-arrows (↑) or the down-arrows (↓) to indicate whether a group is *up* or *down* respectively. Fig. 6(a) shows a pop-up menu associated with the group OH attached to base atom indexed as 'C2'. The menu items are *up, down* or

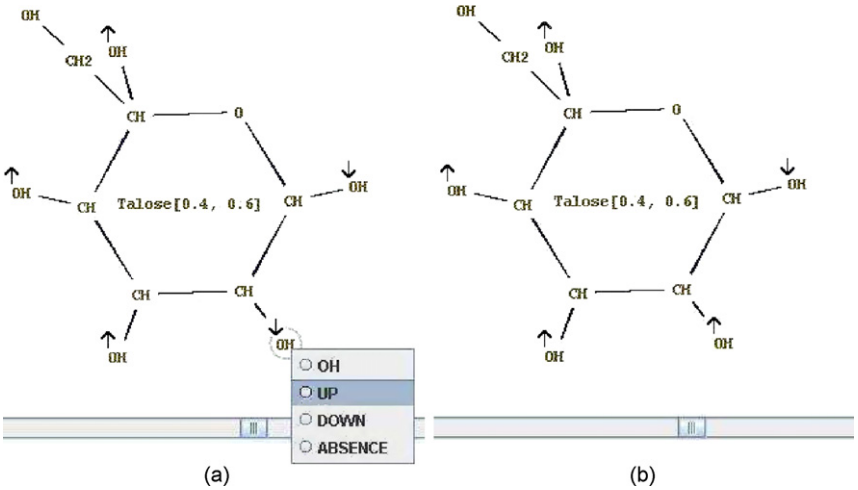


Fig. 6 – Changing the group attached to 'C2' as *up*.

absent. Once a user has chosen *up*, the arrow attached to this group is changed as illustrated in Fig. 6(b).

Each organic compound is translated into a logic formula. After a user inputs the probability interval about the organic compound being a substrate, this piece of information is encoded as a probabilistic formula in the corresponding PLP.

For example, from the XML file in Fig. 5 we can see that for base atoms indexed as C1,..., and C6, there is only one possible group (OH) attachable to them. So when translating this structural knowledge into a logical formula, we simply omit the trivial statements stating the groups is '–OH' group, such as talose is expressed as a logic formula shown below without OH being mentioned explicitly.

$$aldohexose(talose) \bigwedge c1(talose, d) \bigwedge c2(talose, u) \bigwedge c3(talose, u) \bigwedge c4(talose, u) \bigwedge c5(talose, u) \bigwedge c6(talose, p).$$

After a user gives the estimation as [0.4, 0.6], we can get a probabilistic formula in PLP:

$$(sub(talose)|aldohexose(talose) \bigwedge c1(talose, d) \bigwedge c2(talose, u) \bigwedge c3(talose, u) \bigwedge c4(talose, u) \bigwedge c5(talose, u) \bigwedge c6(talose, p))[.4,.6],$$

which states that the probability of talose being a substrate falls in the interval [0.4, 0.6].

4. Application of the framework to substrates prediction

The aim of our system is to provide a very rapid screening for likely ligands (either substrates or inhibitors, depending on the context). It will be particularly useful in situations where a number of similar compounds have been screened experimentally, but information is not available for all possible members of that group of compounds. By providing a simple means to encode existing experimental knowledge and return results within minutes we see this as a valuable addition to initial computational screening approaches. Since our prediction engine uses only existing knowledge, it requires no input concerning the nature of the structure of the target or about the physical nature of the bond strengths in the binding site. Thus it will be applicable to targets (or to unknown proteins identified as a consequence of genome sequencing projects) for which only limited experimental characterization has been carried out. *The key advantage will be that the proposed system can consider all theoretically possible permutations* in a series of ligands and identify, on the basis of limited initial knowledge, which compounds are also likely to be substrates. Consequently we envisage that the prediction tool could become part of the early stages of drug discovery resulting in time (and cost) savings. In addition it may become a valuable tool to those investigating the substrate specificities of newly discovered enzymes.

4.1. *Case Study I: rapid sugar kinase enzymes prediction*

The usefulness of probabilistic logic programs to represent imprecise probabilistic knowledge and harness this knowledge to answer queries can first be demonstrated by an example from biochemistry on the human enzyme galactokinase [11], which uses galactose as a substrate. Galactose has the molecular formula $C_6H_{12}O_6$, but other compounds have the same or similar formula. Since not all possible substrates for the enzyme have been tested, the information regarding this enzyme and its substrates is incomplete. Then the question is: can we predict which will be the substrates for the human enzyme galactokinase based on incomplete and imperfect information? Many factors lead to the information being imperfect including different research laboratories using different criteria for scoring a compound as a substrate and some information is based on galactokinases from other species, so we cannot be certain that substrate specificity is conserved for humans.

Initially probabilities were estimated using experimental data and an element of intuition. Where a particular substrate

Table 1 – The compounds and their probabilities and products to be substrates, obtained from published papers.

Sugar	C1-OH	C2-OH	C3-OH	C4-OH	C5- CH_2 OH	C6-OH	*P* (substrate)	Product	Source
Galactose	D	D	U	U	U	P	1.0	1	[6,7]
Glucose	D	D	U	D	U	P	0.0	0	[6]
2-Deoxygalactose	D	A	U	U	U	P	1.0	0.47	[6]
Fucose	D	D	U	U	U	A	0.0	0	[6]
Talose	D	U	U	U	U	P	[0.4, 0.6]	[0.056, 0.084]	[8,9]
4-Deoxygluocse	D	D	U	A	U	P	[0, 0.5]	[0, 0.021]	[10]
3-Deoxygalactose	D	D	A	D	U	P	[0.6, 0.9]	[0.036, 0.054]	[10]

had been demonstrated experimentally to be a substrate of human galactokinase it was assigned a probability of 1.0. Where there was experimental data indicating that a substrate was not phosphorylated by human galactokinase, a value of 0 was assigned. Compounds which had been shown to be substrates of galactokinase from other species were assigned probabilities between 0 and 1. However, not all substrates are equally good. Therefore a second measure, the *product* was calculated. To calculate this value, the specificity constant k_{cat}/K_m was used, scaled such that the product value with galactose (which is expected to be the best substrate) was equal to 1.0.

Therefore, in Table 1, we have a column representing their probabilities (or intervals) and another column representing their products of the corresponding compounds to be (good) substrates. Column *source* indicates from which published paper this knowledge is obtained. Probabilistic knowledge on compounds in Table 1 is translated to a PLP as shown in Fig. 2.

Using this knowledge base, we can predict the probability of any structure being a substrate for the combination of these six carbons. Twenty-six queries detailed in Table 2 were executed against this PLP and the query results (column Probability, column Product) are presented in Table 2. Below we give our analysis about these query results.

Overall, the predictions appear to over-estimate the probabilities for each possible substrate. For example, given that the fucose (which has the OH group attached to the sixth carbon atom absent) has been shown experimentally not to be a substrate, it is surprising to see compounds which also lack this OH group predicted as having high probabilities as substrates. Of course in compiling the data in Table 1, all the information was weighted equally—for example the presence or absence of the OH group at position 6 was considered of equal worth to the information about the OH at position 2. In fact it is likely that some positions are more important than others in determining substrate specificity. However, in implementing screens such as these, the amount of knowledge to be included will always be a balance between including enough to enable valid predictions, but not so much that the initial knowledge collection and tabulation becomes unreasonably time-consuming.

Despite these limitations, the predictions do appear to have some value in that the ranking of the compounds in terms of their probability of being a substrate seems mostly reasonable and in line with chemical intuition. Ultimately for such a system to be useful to bioscientists, it is this ranking which must be reliable. The most likely use of such a system is to act as a preliminary screen for potential substrates or inhibitors followed by experimental testing of those compounds. Time and expense can be saved if those compounds most likely to be good substrates (or inhibitors) appear at the top of the list and are, therefore, prioritized in the experimental work. Thus the absolute values of the predicted probabilities are less important than the rank order of the compounds.

4.2. *Case Study II: substrate prediction for NQO1*

NAD(H)-quinone oxidoreductase 1 (NQO1) is a broad specificity enzyme which catalyses the reduction of a range of aromatic compounds. It was chosen for the second case study as a large variety of different compounds (including quinones, nitroaromatics and benzimidazoles) have been tested as substrates. In contrast to the galactokinase study, the chemical diversity of the known substrates is wider leading to a greater number of variables to consider.

Two of the many compounds which have been tested experimentally as substrates for NQO1. Shown in Fig. 7(a) is a quinone compound, benzo-1,4-quinone and in Fig. 7(b) a nitroaromatic compound 1,4-dinitrobenzene. Representing these compounds in tabular form required assigning each position in the six-membered ring a letter descriptor from A to F. For each molecule, the most oxidised substituent was placed at the top of the structural representation and designated A. Positions B through F were then defined by moving round the ring sequentially in an anti-clockwise fashion. In these initial studies we concentrated on six membered rings substituted with ketone, methyl and nitro groups.

In this initial case study, knowledge was collected from a limited number of papers [12,13] which described the activity of the enzyme towards a number of structurally related compounds (Table 3). Probabilities were derived from published data in these papers on specificity constants in which the error in the experimental determination was used to define the range of values. The compound given in Table 3 row 5 is shown in our system in Fig. 8.

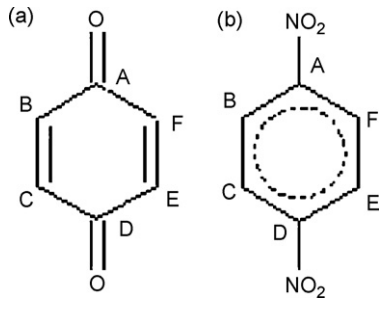


Fig. 7 – Examples of NAD(H)-quinone oxidoreductase 1 (NQO1) substrates.

Table 2 – The probabilities and products of some compounds being a substrate by querying on the PLP.

Sugar	C1-OH	C2-OH	C3-OH	C4-OH	C5- CH_2 OH	C6-OH	P (substrate)	Product
2dAll	D	A	D	D	U	P	0.6529	0.4611
2dGlc	D	A	U	D	U	P	0.6154	0.3939
2dGul	D	A	D	U	U	P	0.6694	0.5000
I	D	A	A	D	U	P	0.5869	0.4083
II	D	A	A	U	U	P	0.6676	0.5376
2,3,4d	D	A	A	A	U	P	0.5509	0.4721
3dAll	D	D	A	D	U	P	0.6003	0.1138
3dMan	D	U	A	D	U	P	0.5539	0.5000
3dTal	D	U	A	U	U	P	0.5636	0.4282
III	D	D	A	A	U	P	0.5321	0.3503
IV	D	U	A	A	U	P	0.5134	0.4785
4dAll	D	D	D	A	U	P	0.5314	0.4611
4dMan	D	U	U	A	U	P	0.4706	0.4282
V	D	A	D	D	U	A	0.5463	0.4811
VI	D	A	U	D	U	A	0.5481	0.4514
VII	D	A	D	U	U	A	0.5481	0.5000
VIII	D	A	A	D	U	A	0.5703	0.4572
IX	D	A	A	U	U	A	0.5682	0.5020
X	D	A	A	A	U	A	0.5233	0.4814
XI	D	D	A	D	U	A	0.5451	0.3518
XII	D	U	A	D	U	A	0.5234	0.5000
XIII	D	U	A	U	U	A	0.5278	0.4670
XIV	D	D	A	A	U	A	0.5146	0.4179
XV	D	U	A	A	U	A	0.5064	0.4895
XVI	D	D	D	A	U	A	0.5144	0.4811
XVIII	D	U	U	A	U	A	0.4879	0.4670

Table 3 – The compounds and their probability intervals, obtained from published papers.

A	B	C	D	E	F	Probability
NO_2	H	H	H	H	H	[0, 0]
NO_2	H	NO_2	H	H	H	[0, 0]
NO_2	H	H	CHO	H	H	[0, 0]
NO_2	NO_2	H	H	H	H	[0, 0]
NO_2	H	H	NO_2	H	H	[0, 0]
O	H	H	O	H	H	[0.20, 0.28]
O	CH_3	H	O	H	H	[0.17, 0.31]
O	CH_3	H	O	CH_3	H	[0.19, 0.33]
O	CH_3	CH_3	O	CH_3	H	[0.20, 0.28]

When used to make predictions about unknown compounds (Table 4), the results were broadly similar to those seen in Case Study I. Table 4 gives the summary of 16 queries based on the probabilistic knowledge given in Table 3. There appeared to be a tendency to over-estimate probabilities (especially for compounds closely related in structure to those with low, or zero, experimentally determined activity). Nevertheless, if these compounds are excluded the rank order of the remaining ones appears sensible. Although further work is required, this method has the potential to provide a valuable, additional tool for the rapid prediction of substrates and inhibitors of enzymes.

5. Related work and conclusion

From IT system's perspective, a few systems have been implemented to modeling and querying probabilistic knowledge, for example, SPIRIT [14] and PIT [15].

In order to manage imprecise probabilistic reasoning, an expert system shell, SPIRIT, was implemented which uses the principle of maximum entropy to avoid the request of precise probability distributions. Knowledge acquisition is performed by specifying probabilistic facts and rules on discrete variables in an extended propositional logic syntax. The shell generates the unique probability distribution which respects all facts and rules and maximizes entropy. After creating this distribution the shell is ready for answering simple and complex queries. System PIT (Probability Induction Tool) was imple-

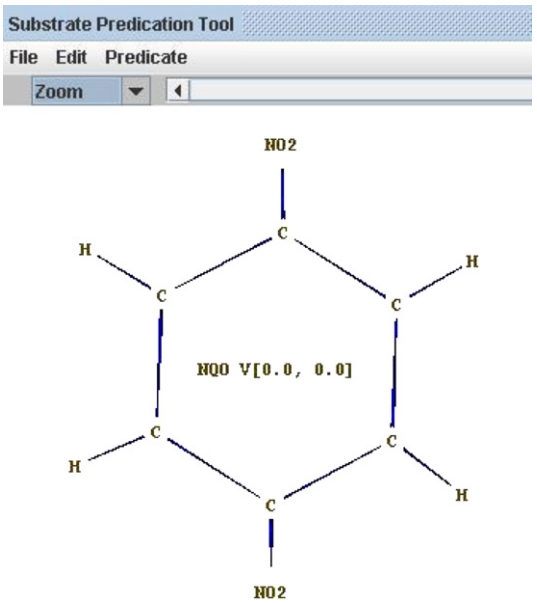


Fig. 8 – One example of NQO1 substrate.

Table 4 – The predictions for some compounds.

A	B	C	D	E	F	Probability
NO_2	H	H	H	NO_2	H	0.0000
NO_2	H	H	NO_2	CH_3	H	0.3194
NO_2	H	H	CHO	CH_3	H	0.3194
NO_2	H	H	O	CH_3	H	0.3294
NO_2	H	NO_2	H	CH_3	H	0.3217
NO_2	H	NO_2	NO_2	H	H	0.1949
NO_2	H	NO_2	O	H	H	0.2235
NO_2	NO_2	H	O	H	H	0.2172
O	H	H	H	NO_2	H	0.2949
O	H	H	NO_2	CH_3	H	0.3917
O	H	H	CHO	CH_3	H	0.3197
O	H	H	O	CH_3	H	0.3629
O	H	NO_2	H	CH_3	H	0.4000
O	H	NO_2	NO_2	H	H	0.3612
O	H	NO_2	O	H	H	0.3477
O	NO_2	H	O	H	H	0.3338

mented based on propositional logic, the probability calculus and the concept of model-quantification. The task of PIT is to deliver decisions under incomplete knowledge but to keep the necessary additional assumptions as minimal as possible.

In contrast, our system deploys the reasoning mechanism in conditional probabilistic logic programming which is based on first-order logics, rather than propositional logics. In addition, we provided a tailored graphical user interface for bioscientists to input knowledge and to query against a knowledge base.

Our system differs from most others used for the screening of potential ligands. Typically, such systems attempt to create computational representations of the physical environment in the enzyme's active site. Therefore, considerable information is required to contribute to the so-called force fields which contain data about the strengths of different types of bonds and interactions. The simplest force fields represent atoms as balls and bonds as springs which obey basic physical laws (e.g. Hooke's law of stretching). However, modern forcefields (e.g. CHARMM, GROMOS and AMBER [16–18]) also incorporate quantum mechanical parameters in order to provide a more realistic model of molecular interactions at the atomic level. Typical investigations involve docking of ligands into active sites, followed by minimization of the calculated energy of interaction. Thus the process involves the utilization of a series of algorithms and programs, often selected by the user on the basis of previous experience. In contrast, the system described here works only with existing knowledge about the ligands. Consequently, it is potentially applicable to any target-based drug discovery programme where some experimental information is available on a range of compounds with related structures. This includes those situations where there is limited functional or structural information about the target. Indeed it may even be applicable when the precise, biochemical nature of the target is not known but where there is considerable, quantitative information about biological responses to a range of compounds (e.g. microbial cell death in the presence of potential, novel antibiotics).

In this paper, we introduced and detailed a rapid substrate prediction tool developed based on reasoning with imprecise probabilistic knowledge. To make the use of tool easier to bioscientists, we developed a graphical user interface which can display the visual structure of a compound. To facilitate the management of the collection of base compound structures and the exchange of this dataset with other systems, we represented the information about base compound structures using XML files—a popular structure for exchanging data with web-based applications.

Two case studies were used to demonstrate the initial results of predictions based on two extracted knowledge bases. Due to the fact that the two knowledge bases are relatively small in size, the prediction results cannot at the moment determine which compounds are extremely likely to be substrates. We anticipate that with more knowledge being added to these two bases, the prediction accuracy will increase.

Conflict of interest

The authors declare that they have no conflict of interests.

Acknowledgement

This work is funded by the EPSRC project with reference number: EP/D070864/1.

REFERENCES

[1] N. Fuhr, Probabilistic datalog: implementing logical information retrieval for advanced applications, JASIS 51 (2) (2000) 95–110.
[2] C. Baral, M. Hunsaker, Using the probabilistic logic programming language p-log for causal and counterfactual reasoning and nonnaive conditioning, in: Proceedings of International Joint Conference on Artificial Intelligence, 2007, pp. 243–249.
[3] L.D. Raedt, A. Kimmig, H. Toivonen, Problog: a probabilistic prolog and its application in link discovery, in: Proceedings of International Joint Conference on Artificial Intelligence, 2007, pp. 2462–2467.
[4] G. Kern-Isberner, T. Lukasiewicz, Combining probabilistic logic programming with the power of maximum entropy, Artificial Intelligence 157 (1–2) (2004) 139–202.
[5] T. Lukasiewicz, Probabilistic logic programming, in: Proceeding of European Conference on Artificial Intelligence (ECAI), 1998, pp. 388–392.
[6] D.J. Timson, R.J. Reece, Sugar recognition by human galactokinase, BMC Biochemistry 4 (2003) 16, http://www.biomedcentral.com/1471–2091/4/16.
[7] D.J. Timson, R.J. Reece, Functional analysis of disease-causing mutations in human galactokinase, European Journal of Biochemistry 270 (8) (2003) 1767–1774.
[8] C.A. Sellick, R.J. Reece, Contribution of amino acid side chains to sugar binding specificity in a galactokinase, Gal1p, and a transcriptional inducer, Gal3p, Journal of Biological Chemistry 281 (25) (2006) 17150–17155.
[9] J. Yang, L. Liu, J.S. Thorson, Structure-based enhancement of the first anomeric glucokinase, ChemBioChem 5 (7) (2004) 992–996.
[10] J. Yang, X. Fu, Q. Jia, J. Shen, J.B. Biggins, J. Jiang, J. Zhao, J.J. Schmidt, P.G. Wang, J.S. Thorson, Studies on the substrate specificity of Escherichia coli galactokinase, Organic Letters 5 (13) (2003) 2223–2226.

[11] H.M. Holden, J.B. Thoden, D.J. Timson, R.J. Reece, Galactokinase: structure, function and role in type II galactosemia, Cellular and Molecular Life Sciences (CMLS) 61 (2004) 2471–2484.
[12] N. Cenas, A. Nemeikaite-Ceniene, E. Sergediene, H. Nivinskas, Z. Anusevicius, J. Sarlauskas, Quantitative structure–activity relationships in enzymatic single-electron reduction of nitroaromatic explosives: implications for their cytotoxicity, Biochimica et Biophysica Acta 1528 (1) (2001) 31–38.
[13] Z. Anusevicius, J. Sarlauskas, N. Cenas, Two-electron reduction of quinones by rat liver NAD(P)H:quinone oxidoreductase: quantitative structure–activity relationships, Archives of Biochemistry and Biophysics 404 (2) (2001) 254–262.
[14] W. Rödder, E. Reucher, F. Kulmann, Features of the expert-system-shell spirit, Logic Journal of the IGPL 14 (3) (2006) 483–500.
[15] M. Schramm, V. Fischer, Probabilistic reasoning with maximum entropy—the system pit (system description), www.pit-systems.de/Pit/P_Lit/P_Download/SF97.ps, 1997.
[16] D.A. Case, T.E. Cheatham, T. Darden, H. Gohlke, R. Luo, K.M. Merz, A. Onufriev, C. Simmerling, B. Wang, R.J. Woods Jr., The Amber biomolecular simulation programs, Journal of Computational Chemistry 26 (2005) 1668–1688.
[17] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, CHARMM: a program for macromolecular energy, minimization, and dynamics calculations, Journal of Computational Chemistry 4 (1983) 187–217.
[18] C. Oostenbrink, A. Villa, A.E. Mark, W.F. van Gunsteren, A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6, Journal of Computational Chemistry 25 (2004) 1656–1676.