ELSEVIER

Optimizing the use of open-source software applications in drug discovery

Werner J. Geldenhuys[1], Kevin E. Gaasch[2], Mark Watson[2], David D. Allen[1] and Cornelis J. Van der Schyf[1,3]

[1]Department of Pharmaceutical Sciences, School of Pharmacy, Texas Tech University Health Sciences Center, Amarillo, TX, USA
[2]West Texas A&M University, Canyon, TX, USA
[3]Pharmaceutical Chemistry, School of Pharmacy, North-West University, Potchefstroom, South Africa

Drug discovery is a time consuming and costly process. Recently, a trend towards the use of *in silico* computational chemistry and molecular modeling for computer-aided drug design has gained significant momentum. This review investigates the application of free and/or open-source software in the drug discovery process. Among the reviewed software programs are applications programmed in JAVA, Perl and Python, as well as resources including software libraries. These programs might be useful for cheminformatics approaches to drug discovery, including QSAR studies, energy minimization and docking studies in drug design endeavors. Furthermore, this review explores options for integrating available computer modeling open-source software applications in drug discovery programs.

To bring a new drug to the market is very costly, with the current price tag approximating US$800 million, according to data reported in a recent study [1]. Therefore, it is not surprising that pharmaceutical companies are seeking ways to optimize costs associated with R&D, with the goal of increasing profit margins. One method that was quickly adopted by industry was the use of combinatorial chemistry and HTS. In HTS, large libraries of compounds are screened against drug targets to identify lead compounds that can modulate a particular outcome. However, setting up a combinatorial chemistry program and HTS is costly and not able to address the specific needs of many biological (drug target) systems [2,3]. Additionally, compounds identified in such screenings are not always amenable to further medicinal chemistry development, with poor ADME (absorption, distribution metabolism and elimination) properties [4]. Although these methods have increased the rate at which lead compounds can be identified, there has not been a commensurate increase in the rate of introduction of new chemical entities (NCE) into the world drug market [5].

As an attractive alternative, *in silico* methods show promise in identifying new lead compounds faster and at a fraction of the cost of combinatorial approaches and HTS. The addition of computer-aided drug design technologies to the R&D approaches of a company, could lead to a reduction in the cost of drug design and development by up to 50% [6,7]. These *in silico* methods encompass a wide terrain, including: (i) docking studies, where a ligand or drug is computationally studied during the binding to a particular target protein; (ii) cheminformatics, where activity and structure are correlated using statistical means; or (iii) bioinformatics, where drug targets are derived from genomic data. The use of *in silico* drug design has led to the discovery of indinavir, the HIV protease inhibitor [8], and the identification of haloperidol as a lead compound in a structure-based design study for nonpeptide inhibitors of HIV [9].

The software that is available for computer-aided drug design and development originates from different sources. These include commercial companies, academic institutions, open-source software or in-house development. Each of these sources has its pros and cons, and the appropriate choice varies for institutions that use the software. These software packages also differ in terms of cost, functionality and efficacy [10,11], and automation [12]. Table 1 gives a summary of some considerations that a company or scientist might use when evaluating a new software package.

Corresponding author: Geldenhuys, W. J. (werner.geldenhuys@ttuhsc.edu).

1359-6446/06/$ – see front matter PII: S1359-6446(05)03692-5

TABLE 1

Summary of possible considerations when selecting an *in silico* drug-design package

	Vendor of software			
	Academia	**Commercial**	**Open-source**	**In-house**
Price	Free to expensive	Expensive	Free (usually to private and academic; commercial enterprises might have to pay fees)	Expensive – takes up time of specialized personnel
Validated and/or tested	Usually tested – with some bugs needing to be reported	Tested – few bugs	Responsibility of testing falls upon developers and user community, hence, is dependent on product following	Responsibility by internal team
Support	Varies from product to product	Timely reply	Dependent on developer and user following; some open-source developers contract their services for support	Very good, internal group
User-friendly	Varies	Yes	Seldom	Depends
Adaptability	Sometimes limited	Support will help with additions to programs	Very adaptable	Very adaptable and greatest degree of flexibility to meet company demands

Free and open-source software

In the past few years, free and open-source software has gained in popularity with an increasing number of websites dedicated to such packages, one example being sourceforge.net. Table 2 gives a list of programs that are under the open-source license structure. Because projects change constantly, this list is not intended to be exhaustive and the internet addresses might change with time.

In general, open-source refers to any program that has the source code available for use or modification as users or other developers see fit (historically, the developers of proprietary software have generally not made the source code available). Open-source software is usually developed as a public collaboration and made available freely (www.opensource.org), although 'free' can sometimes refer only to academic institutions, whereas commercial enterprises might be required to pay a fee.

Recently, DeLano [13] discussed the use of open-source software in the drug discovery environment. The use of open-source software has a number of advantages that make it attractive, especially to the academic scientist. Historically, academic scientists have had less funding to invest in their projects. Additionally, commercially available drug-design software packages usually come with an expensive license fee, renewable every year. Some of the advantages for using open-source and free drug discovery software discussed by DeLano [13] include the ability to download a program directly and immediately from the internet (rapid implementation), no license fees and flexibility at a lower cost with options to customize the software for a particular project. These features offer an obviously attractive way for programmers to share and advance ideas, in addition to not having to start from scratch when producing new software programs [14].

Although open-source software programs appear very attractive because they allow a programmer a head start on a project and are usually free, there are some pitfalls that users and developers need to be aware of [15]. Software programs are not always well written or their use well documented, which might present a problem for the average end-user. Also, especially in the field of chemistry, commercialization has been a driving force, sometimes making it difficult to convince experts in a field to contribute in their 'spare' time to these open-source projects. Open-source programs, as seen on a depository such as sourceforge.net, are usually developed by a few regular contributors, with other contributors adding to the program on a less frequent basis [15] or by graduate students majoring in computational or computer science as part of their thesis work. Consequently, many of these programs do not come with easy installation and most of them need to be compiled by the programming language-specific compiler (C++, Fortran, JAVA) or run from the command line. Additionally, most of these programs are written for Linux or SGI platforms, because programming for the Microsoft® environment is a tedious and often difficult task, even for simple projects (www.codeproject.com/library/win32easy.asp). This is probably a contributing reason to why many cheminformatics and bioinformatics scientific software programs have poorly written graphical user interfaces (GUIs) that are not user friendly. In the absence of well-written user manuals and clear examples of how to use the program, the lack of user friendliness can result in a bench scientist spending more time trying to install or manage the program, rather than using it. Such unproductive time use is clearly not the reason why *in silico* methods are used in the first place – that is to save time, thereby saving money. Some software programs are set up to run from what is referred to as command line, where, in the absence of a GUI, commands are given to the program by typing into a command line editor. This type of interaction with a software program is preferred by some users because it has the advantage of speeding up processes that require a few steps to complete, especially when used for large dataset analysis.

As an alternative to the single platform environment and the problems it creates, some software programmers have attempted to employ web-based technologies. JAVA applets, for instance, are loaded and run from the internet on any compatible web browser. JAVA can also operate outside of the web environment using the JAVA virtual machine (JVM) that allows JAVA programs to run on multiple platforms [16]. The VigyaanCD project supplies a downloadable program that allows the user to boot a computer to run the LINUX environment from a CD-ROM (www.vigyaancd.org). When the computer is then re-booted without the CD, the original operating system is initialized as normal. This approach allows an electronic workbench to run computational chemistry and biology, as well as bioinformatics. Programs available on the CD include Arka/GP, Artemis, Bioperl, BLAST, ClustalW/ClustalX, Cn3D, EMBOSS tools, Garlic, Glimmer, GROMACS, Ghemical, GNU R, Gnuplot, GIMP, ImageMagick, Jmol, MPQC, MUMer, NJPlot, Open Babel, Octave, PSI3, PyMOL, Ramachandran plot viewer, Rasmol, Raster3D, Seaview, TINKER, XDrawChem, Xmgr and Xfig, among others.

TABLE 2

Examples of free and/or open-source software packages for computational and molecular modeling, available for free download from the internet

Application	Program name	Web sites
Visualization	Rasmol	www.openrasmol.org
	MolVis	http://molvis.sdsc.edu/visres
	PyMol	http://pymol.sourceforge.net
	DeepView	http://us.expasy.org/spdbv/
	JMol	http://jmol.sourceforge.net/
	gOpenMol	www.csc.fi/gopenmol/
	AstexViewer	www.astex-therapeutics.com
Docking	ArgusDock	www.arguslab.com
	DOCK	http://dock.compbio.ucsf.edu/
	FRED	www.eyesopen.com/
	eHITS	www.simbiosys.ca/
	AutoDock	www.scripps.edu/
	FTDock	www.bmm.icnet.uk/docking/ftdock.html
Energy minimization	GAMESS	www.uiowa.edu/~ghemical/gtk-gamess.shtml
	Ghemical	www.uku.fi/~thassine/ghemical/
	PS13	www.psicode.org/
	TINKER	http://dasher.wustl.edu/tinker
QSAR descriptors	SoMFA	http://bellatrix.pcl.ox.ac.uk/
	GRID	www.moldiscovery.com/
	E-Dragon 1.0	http://146.107.217.178/lab/edragon/
	ALOGPS 2.1	http://146.107.217.178/lab/alogps/
	Marvin Beans	www.chemaxon.com/
Chemical drawing	ACD/labs ChemSketch	www.acdlabs.com/
	ISISDraw	www.mdli.com/
	XDrawChem	http://xdrawchem.sourceforge.net/
	JME Editor	www.molinspiration.com/jme/
Software libraries	Chemical Development Kit	http://almost.cubic.uni-koeln.de/cdk/
	Molecular Modeling Toolkit	http://starship.python.net/crew/hinsen/MMTK/
	PerlMol	www.perlmol.org
	JOELib	www-ra.informatik.uni-tuebingen.de/software/joelib/
	OpenBabel	http://openbabel.sourceforge.net

In lieu of a commercial product, a bench scientist might elect to download a number of free programs to obtain results similar to those that would have been achieved with the commercial software. A good example of this would be to compare a commercial product such as SYBYL (www.tripos.com) to free and/or open-source software. For instance, the program XDrawChem might be useful for drawing a chemical structure for publication. For energy minimization, a program such as Ghemical can be used, or PS13 for quantum mechanical calculations. If proteins are used, a program such as PyMol can be used to identify ligand binding pockets, together with the DeepView PDB viewer to investigate the amino acid sequences of the protein. To transfer files between programs, Open Babel might be useful or even required to interconvert the file formats.

Web-based programs

The internet is also a source of web-based tools that focus on computational chemistry. For instance, JAVA applets have made it possible to create a web-based interface that can be used by a bench scientist without the need to install a program or run a specific operating system on a desktop computer. Furthermore, applets have the added advantage of allowing rapid upgrades or changes to be made to the software program, allowing immediate deployment through updating a server that hosts the website [17]. An example of such a website is the log *P* calculator from Interactive Analysis (www.logp.com). Using the interactive interface, an organic compound can be drawn, a text version of an MDL mol file [18] pasted in a box or a SMILES code inserted for the structure, after which log *P* for the compound will be calculated upon submitting the query.

Another example is the open-source software molecular structure viewer JMol (Table 2). This program runs as an interactive web-browser JAVA applet, using JAVA built into the most popularly used web-browsers. A major advantage of JMol is that this visualizer can be incorporated into in-house (program produced exclusively for a user's need) programs, serving as a packaged application programming interface (API) or library for other programs without a user interface. The University of Utah has set up a virtual laboratory,

Computational Science and Engineering online (www.cse-online.net), which allows users to connect to a Unix or Linux server, submitting jobs that include molecular mechanics, quantum chemical calculations and biomolecular interfaces for viewing protein databank (PDB) files. This option is free for users using Unix or Linux servers online. JOELib (www-ra.informatik.uni-tuebingen.de/software/joelib/index.html) is another library written in JAVA, for computational and cheminformatics software, which is also able to calculate QSAR parameters such as log *P*, Kier shape, molecular refractivity and Gasteiger-Marsili atom charges.

The internet has also seen an increase in the use of web services for drug discovery (for recent reviews see [19] and [20]). Web-based tools have some advantages for the communication of chemical information [19,21], one of the most important of these being the development of user-friendly interfaces. Bench chemists are sometimes dissuaded from becoming more involved with molecular modeling and cheminformatics because of the requirement to learn commands in UNIX, a system on which most commercial programs are developed [19]. Additionally, using web-based tools, a user is able to test the software and evaluate its usefulness. This might include whole software suites or only individual algorithms [19]. The use of web-based cheminformatics tools offers great potential when deployed as an integrated part of a pharmaceutical company's intranet, one example being the program developed by Novartis [19,21]. This service offers the bench chemist the ability to calculate molecular properties of compounds, such as polar surface area and log *P*. Because of intellectual property considerations, most industrial scientists will likely use services provided on their own company's intranet, as opposed to the free services available over the (open) internet, which are mostly used by academic scientists [19].

Docking validation and datasets

The accurate prediction of binding between a ligand and protein is a difficult and challenging area of computational chemistry [22]. The use of reference compounds in biological assays is a common occurrence in scientific literature because of the amount and quality of information that can be extracted from such an exercise. Using a reference compound allows other researchers to assess the robustness of their assay *vis-à-vis* a previously published article and acts as a measure to compare the activity of new compounds with that of the reference. The use of validation calculations for open-source software might not always be easily accessible, requiring the bench chemist to test the program with a set reference project that is similar in nature. For example, in the case of docking studies, few datasets are available to the general scientific community with binding affinities for a particular target [23]. Although the pharmaceutical industry has acquired large datasets (due to HTS), companies closely guard their intellectual property, and their in-house datasets are rarely available for scientific publication.

In docking programs, there is a dearth of known standards and datasets of compounds that bind into different protein receptors or enzymes and that can be used to validate a particular program. Additionally, docking programs do not always function well in identifying hit compounds that were not included in a training set, and some scoring functions render results similar to a random selection when ranking the fit of compounds in a protein [24]. The output of docking studies is known as scoring function, with accuracy measured by the difference between experimental and calculated binding energies or by attempts to limit the root-mean-square (RMS) fit or deviation to less than 2.0 between the docked pose and the experimental crystallized compound [24]. For instance, to evaluate the free docking program ArgusDock (www.arguslab.com) [25], the docking scores of 786 structures from the PDBbind database [26,27] were scored and compared with their experimentally derived binding energies. A dataset that stands out is the set of protein–ligand complexes used by the GOLD docking program. To validate the GOLD docking program, 100 PDB protein–ligand complexes were used and the docking solutions investigated. Another useful compilation is the PBD files used by Cozzini *et al.* [22], a set of 210 ligand–protein complexes, with 3D structures and binding energies already known from the literature. Using any of these sets to validate another program and describing the method of validation in scientific literature would increase the comparability and effectiveness of one docking program with another, thereby assisting bench scientists to select a particular program for their individual needs more easily.

Other available datasets are those from the QSAR society (www.qsar.org) and from Cheminformatics.org (www.cheminformatics.org). Unfortunately, there is a limited number of datasets and little information regarding the training and validation used by previous researchers. Tetko *et al.* [14] suggested the use of SMILES or .sdf files on a website to promote the calculation of additional parameters by other drug discovery scientists. The self-organizing molecular field analysis (SoMFA) test set, which represents the steroid set used to construct the first comparative molecular field analysis (CoMFA), can be downloaded from the Richards group's web site (http://bellatrix.pcl.ox.ac.uk). This information facilitates a more-rapid evaluation of the SoMFA program.

Property calculation programs

QSARs are based on the assumption that the biological activity of a compound is related to its molecular properties. The origins of QSAR are rooted in the works of Hansch *et al.* [28] and Free *et al.* [29]. Cheminformatics has progressed significantly since then, with advances in computer hardware and software to calculate molecular properties, with the aim of identifying compounds that might become useful drugs [30,31].

The calculation of descriptors that are generally used in traditional 2D-QSAR studies can be done using web-based sites, such as the log *P* calculator from Interactive Analysis or the virtual computational chemistry laboratory (www.vcclab.org) for a more extended parameter set. Using the ALOGP 2.1 program [14], log *P* and log *S* values can be estimated, and with E-Dragon, it is possible to generate 1600 molecular descriptors for compounds. These include topological indices, connectivity indices and geometrical descriptors. The software can be freely downloaded by academic and not-for-profit organizations. Another useful program is Marvin Beans from Chemaxon (www.chemaxon.com). This easy-to-use program is able to calculate a host of molecular descriptors used in QSAR studies, including log *P*, polar surface area, H-bond acceptor and donor numbers, and log *D*. The company Advanced Chemistry Development Laboratories (www.acdlabs.com) offers drawing programs for chemical structures that are also able to calculate descriptors such as molecular weight, molecular formula, molar refractivity, molecular volume, density and polarizability.

CoMFA has become a popular 3D-QSAR technique in the drug discovery process, offering an indirect method to investigate the required properties of a set of compounds for pharmacological activity, where the 3D structure of the target protein is not known. This method was originally developed by Cramer *et al.* [32] using a set of steroid compounds. A probe is used in a 3D lattice to calculate the interaction between electronic and steric properties of a set of compounds. Using the statistical partial least squares technique, these steric and electronic properties are compared with known biological activities and the result shown visually on a computer monitor as field maps of different colors. CoMFA is a proprietary product incorporated into the SYBYL software platform. Recently, a new 3D-QSAR method was developed by Robinson *et al.* [33]. The program SoMFA is similar to CoMFA in that a grid-based methodology is used, but differs in that no probe interaction energies are used but only the intrinsic molecular properties (i.e. molecular shape and electrostatic potentials). The resultant model is shown as a grid with different colors indicating the electrostatic potential or shape surrounding a molecule.

Software libraries for programming

An API or library is a reusable machine language component with functionality that can be utilized within another program. By itself, an API is not a complete program but it contains functionalities needed by many different programs. For instance, a programmer might wish to develop a software program with a rich user interface but also use an API to perform common algorithmic calculations. The API is developed, packaged and delivered separately, but allows a programmer to include and use it in a custom solution.

The chemistry development kit (CDK) is an open-source JAVA (http://java.sun.com) library that can be used when either cheminformatics or bioinformatics application programs are written [34]. The history of CDK originated as a support project for cheminformatics software, aimed at the academic community. The products available are JChemPaint, which is a 2D chemical diagram editor, Seneca, which is a software package aimed at computer-assisted structure elucidation, and NMRShiftDB, an open-content database of organic structures and their NMR data. The source code can be found on sourceforge.net and CDK can be adapted to run under different applications, therefore this library has great flexibility and adaptability. Another open-source library is the molecular modeling toolkit (MMTK) [35], available using the Python programming language (www.python.org). MMKT focuses on molecular simulation techniques, such as rendering proteins. PerlMol (www.perlmol.org) is a library written in Perl, for cheminformatics and computational chemistry applications. The toolkit has modules for molecule visualization, substructure matching and reading and writing files in various formats.

Conclusion

Open-source software has had a significant impact on areas such as bioinformatics but has been slow to impact drug discovery in a similar way. Although a number of free and/or open-source software packages such as QSAR molecular descriptors or docking software are available for drug discovery, they might be inaccessible to the bench chemist because of either a poorly programmed GUI or insufficient literature to validate the program. These deficits might rob the time 'saved' using *in silico* modeling versus bench work. Additionally, a double standard seems to exist for software programs published in scientific journals that could negatively impact bench science – often insufficient information is disclosed in the publication for a 'competent scientist' to reproduce the work [36]. When more use is made of successful free and/or open-source software, especially in the academic community, many more drug discovery projects will benefit and programs with added functionality and user friendliness will, as a result, likely become available to assist such endeavors even further.

Acknowledgments

The authors wish to thank M. Christof Van der Schyf for assistance with comprehensive internet searches.

References

1 DiMasi, J.A. *et al.* (2003) The price of innovation: new estimates of drug development costs. *J. Health Econ.* 22, 151–185
2 Englebienne, P. (2005) High throughput screening: will the past meet the future? *Frontiers Drug Design Discov.* 1, 69–86
3 Geysen, H.M. *et al.* (2003) Combinatorial compound libraries for drug discovery: an ongoing challenge. *Nat. Rev. Drug Discov.* 2, 222–230
4 Bleicher, K.H. *et al.* (2003) Hit and lead generation: beyond high-throughput screening. *Nat. Rev. Drug Discov.* 2, 369–378
5 Wolohan, P.R. *et al.* (2003) Predicting drug pharmacokinetic properties using molecular interaction fields and SIMCA. *J. Comput. Aided Mol. Des.* 17, 65–76
6 McGee, P. (2005) Modeling success with in silico tools. *Drug Discov. Today* 8, 23–28
7 *FDA* (2004) Challenge and opportunities on the critical path to new medical products (http://www.fda.gov/oc/initiatives/criticalpath/whitepaper.html)
8 Wlodawer, A. (2002) Rational approach to AIDS drug design through structural biology. *Annu. Rev. Med.* 53, 595–614
9 DesJarlais, R.L. *et al.* (1990) Structure-based design of nonpeptide inhibitors specific for the human immunodeficiency virus 1 protease. *Proc. Natl. Acad. Sci. U. S. A.* 87, 6644–6648
10 Marchand-Geneste, N. *et al.* (2004) e-Quantum chemistry free resources. *SAR QSAR Environ. Res.* 15, 43–54
11 Carpy, A.J. (2002) WWW small molecule modeling. *SAR QSAR Environ. Res.* 13, 403–408
12 Carpy, A.J. *et al.* (2003) e-Molecular shapes and properties. *SAR QSAR Environ. Res.* 14, 329–337
13 DeLano, W.L. (2005) The case for open-source software in drug discovery. *Drug Discov. Today* 10, 213–217
14 Tetko, I.V. *et al.* (2002) Application of associative neural networks for prediction of lipophilicity in ALOGPS 2.1 program. *J. Chem. Inf. Comput. Sci.* 42, 1136–1145
15 Stahl, M.T. (2005) Open-source software: not quite endsville. *Drug Discov. Today* 10, 219–222
16 Tetko, I.V. (2003) The WWW as a tool to obtain molecular parameters. *Mini Rev. Med. Chem.* 3, 809–820
17 Bembenek, S.D. *et al.* (2004) A web-based cheminformatics system for drug discovery. *Methods Mol. Biol.* 275, 65–84
18 Dalby, A. (1992) Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited. *J. Chem. Inf. Comput. Sci.* 32, 244–255
19 Ertle, P. *et al.* (2004) Web-based cheminformatics tools deployed via corporate Intranets. *Drug Discov. Today: Biosilico* 2, 201–207
20 Curcin, V. *et al.* (2005) Web services in the life sciences. *Drug Discov. Today* 10, 865–871
21 Ertl, P. *et al.* (2003) Web-based cheminformatics and molecular property prediction tools supporting drug design and development at novartis. *SAR QSAR Environ. Res.* 14, 321–328
22 Cozzini, P. *et al.* (2002) Simple, intuitive calculations of free energy of binding for protein-ligand complexes. 1. Models without explicit constrained water. *J. Med. Chem.* 45, 2469–2483
23 Stahl, M. (2000). Modifications in the scoring function in FlexX for virtual screening applications. *Persp. Drug Disc. Des.* 20, 83–98
24 Pick, D. (2004) Novel screening methods in virtual ligand screening. *Methods*

Mol. Biol. 275, 439–448
25 Thompson, M.A. (2004). Poster Presentation: Molecular docking using ArgusLab: An efficient shape-based search algorithm and the AScore scoring function. Fall ACS meeting, Philadelphia.
26 Wang, R. *et al.* (2004) The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *J. Med. Chem.* 47, 2977–2980
27 Wang, R. *et al.* (2005) The PDBbind database: methodologies and updates. *J. Med. Chem.* 48, 4111–4119
28 Hansch, C. *et al.* (1964) σ-p analyis. A method for correlation of biological activity and chemical structure. *J. Am. Chem. Soc.* 86, 1616–1626
29 Free, S.M., Jr *et al.* (1964) A mathematical contribution to structure-activity studies. *J. Med. Chem.* 53, 395–399
30 Esposito, E.X. *et al.* (2004) Methods for applying to the quantitative structure-activity relationship paradigm. *Methods Mol. Biol.* 275, 131–213
31 Brown, F.K. (1998) Cheminformatics: what is it and how does it impact drug discovery. *Ann. Rep. Med. Chem.* 33, 375–384
32 Cramer, R.D. *et al.* (1988) Comparative molecular field analysis (CoMFA). 1. Effect of shape on binding of steroids to carrier proteins. *J. Am. Chem. Soc.* 110, 5959–5967
33 Robinson, D.D. *et al.* (1999) Self-organizing molecular field analysis: a tool for structure-activity studies. *J. Med. Chem.* 42, 573–583
34 Steinbeck, C. *et al.* (2003) The Chemistry Development Kit (CDK): an open-source Java library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* 43, 493–500
35 Hinsen, K. (2000) The molecular modeling toolkit: A new approach to molecular simulations. *J. Comp. Chem* 21, 79–85
36 Barcza, S. (1991) A case for sharing. *J. Comput. Aided Mol. Des.* 5, 509–510