SOMA − Workflow for Small Molecule Property Calculations on a Multiplatform Computing Grid

Pekka T. Lehtovuori* and Tommi H. Nyrönen*

The Finnish IT Center for Science CSC, P.O. Box 405, FI-02101 Espoo, Finland

Received September 9, 2005

We present the concept of the SOMA workflow developed at the Finnish IT Center for Science CSC. The SOMA workflow unites multiplatform UNIX/LINUX computing resources and third-party software for calculating molecular structure and properties. The presented workflow components consist of the computing program XML descriptions, the core workflow program *Grape*, the toolkit for parsing program input and output, and the extranet interface. The program *Grape* and the developed XML descriptions of scientific programs allow researchers to link molecular modeling software into highly sophisticated computational workflows. SOMA collects the calculated data produced by the workflow and stores the computed information in the Chemical Markup Language (CML) format. The extranet interface is used for user authentication, building of the program interfaces and the workflows, and for sorting, filtering, and visualizing the results.

INTRODUCTION

After a useful combination of computing tools has been figured out by a scientist, mechanisms are needed in order to share the strategy with colleagues.[1−4] The traditional way is to present the workflow and explain what the scripts used to run the series of programs in the computing environment do.[3,5] If colleagues find the sequence and the scripts interesting, based on the textual or verbal description, probably the only way of sharing the workflow is by copying the scripts and reconstructing the workflow. This procedure, however, is technologically inefficient and cumbersome.

Overcoming the technical issues related to better integration of molecular modeling and information management is tackled in many laboratories and companies worldwide.[2−4,6−10] The challenge of integration lies in the fact that chemical knowledge and data from molecular modeling are still presented in a plethora of data formats.[11] Chemical Markup Language (CML)[12] is a good effort toward standardization. A data format that becomes a standard way to present molecular information and the metadata for the molecule will greatly benefit the cheminformatics community.

Standards for presenting molecular information would improve the cross-talk between software made by various providers. In bioinformatics workflows of operations carried out to, e.g., sequence, gene expression, and structural information are a subject of much attention.[1,13−15] Similarly in cheminformatics, if software could be linked more easily together more versatile molecular information could be produced by using the best combination of software and algorithms to do the computational experiments. However, only a few scientists can construct even conceptually simple computational workflows, e.g., how one can transfer data from a chemical database to a 2D-3D converter software, pass on the information to a docking program, and run a batch queue on a UNIX server. Once recognized, processes of this type can and should be more automatic and easier to do.[7,8] Clearly there is a call for technology that will further improve the cross-talk between cheminformatics software.

Scientific service providers have to be able to offer scientifically relevant and useful services. It is clear that provision of raw computing capacity with/without software easily leads to many scientists doing redundant work. Services are typically composed of a set of frequently updated software installed in a computing environment that is also constantly evolving. As the system becomes more complex, the resources from supporting research are quickly transformed into an effort to orchestrate a functioning cooperation of hardware and software.

These technical problems constitute a relevant challenge for the molecular modeling research community. Not being able to share knowledge efficiently inhibits a wider use of simulations and computing as an integral tool in research. Scientists have an aptitude for solving problems concerning, e.g., chemistry or biosciences, but seldom possess the knowledge (or time) to venture deeply into computer techniques that could help in the task.

In this work we describe the concepts of the SOMA workflow that we have created for the building of molecular modeling workflows used in computer-aided drug design. We have focused our work to help the researchers to be able to use computer simulations as an aid in their work, rather than spending time solving problems related to the technical complexity of the tools or the computer environment where these tools are used. We present tools that alleviate the technical problems involved in the efficient use of software from different providers installed on a multiplatform UNIX/LINUX environment. The SOMA workflow consists of an interface, program descriptions, core workflow program, and a toolkit of utility programs, which help in the attachment of new applications to SOMA. SOMA allows users to link together program input and output into a workflow and controls the collection of calculated data produced by the

* Corresponding author phone: +358-9-4572070; e-mail: pekka.lehtovuori@csc.fi (P.L.) or phone: +358-9-4572235; e-mail: tommi.nyronen@csc.fi (T.N.).

10.1021/ci050388n CCC: $33.50
Published on Web 01/11/2006

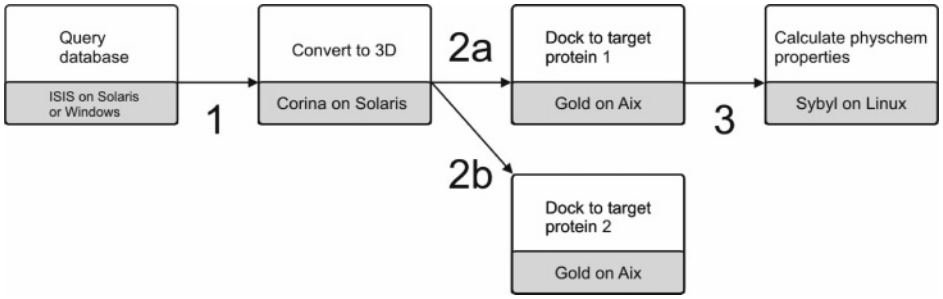

Figure 1. A use case of molecular modeling in drug discovery. The user first queries a chemical supplier database using a set of criteria and filters that can be derived, e.g., from the known ligands or the target receptor structure. Having reached a suitable number of compounds for further experiments, the 2D structures retrieved from the database to 3D structures (**1**) are converted and are docked to the target receptors. After two docking simulations (**2a**, **2b**) a set of physicochemical descriptors of the compounds (**3**) are calculated and their pharmacokinetic properties are assessed.

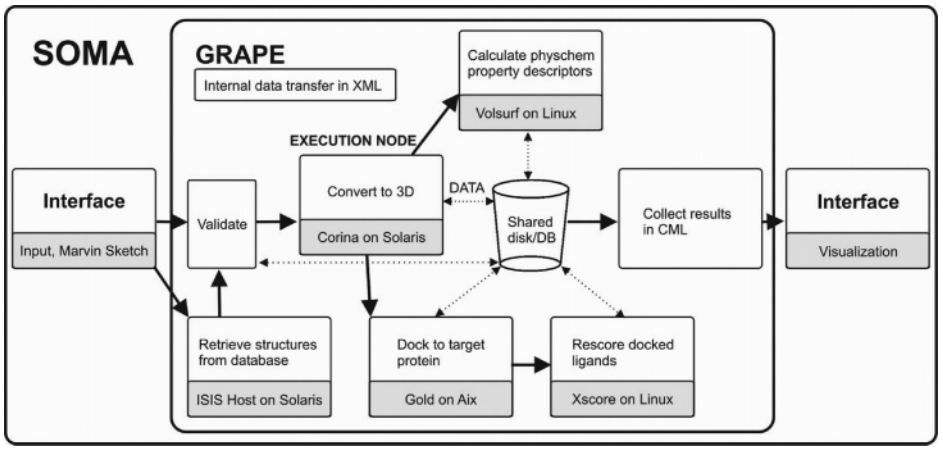

Figure 2. An exemplary execution graph of the SOMA workflow with arrows depicting data transfer. The execution graph consists of several execution nodes which are launched by the program *Grape*.

workflow. The interface provides a secure connection to the service at the Finnish IT Center for Science CSC, authenticates users and helps in building the workflows, and sorts, filters, and visualizes the results.

METHODS AND RESULTS

Overview. The SOMA workflow is made for overcoming technical problems in molecular structure and property calculations encountered in drug discovery. It consists of third-party scientific applications that are linked together and accessed through a Web browser forming a scientist-intuitive computing and data management environment for small molecule data. SOMA aims to make the computing software more user-friendly for the scientists, to mask the technical complexity of multiplatform computing environments, and to automate repeating steps in molecular modeling.

Use Case. A question from a scientist optimizing molecular structures and properties might be the following: "I would like to know if commercially available analogues of a ligand are available and likely to bind to my target protein. In addition I need to know if it is likely that analogues will bind to a certain closely related protein, and how lipophilicity varies among different analogue structures.".

This rather simple workflow (Figure 1) involves technical steps, of which some are nontrivial and laborious. The exported 2D structures from the chemical database may need to be transferred to the server where the conversion program is installed. After running conversion, the user must check that the molecular structures are suitable input for the docking program. Already at this stage, technical problems may arise. The 2D-3D conversion program and the docking program that need to connect in the desired workflow are possibly not designed to work together. The user must also configure and commit the preferred docking program twice as the ligand set is being docked to two receptors. It is probable that docking requires more computing power than the other steps, thus files might have to be moved to a more powerful computing server. After making the docking studies, a way is needed to associate the results of the docking with the calculated physicochemical properties, as it is not efficient to do the same calculation twice on the same set of ligands. Moreover, many third-party programs are platform dependent, thus quite a bit of uploading and downloading of data between servers may end up being done before having successfully accomplished all required steps and collecting the produced data together. In addition to knowing how the necessary programs are run, it is necessary to know how each of the operating systems batch queue parameters, etc. are set. Challenges lay both in the data transfer between scientific software that are not designed to work together and in the assembly, analysis, and visualization of the accumulated data. All these steps can be automated to a great extent in the SOMA workflow.

Example of Use. The service user authenticates to the SOMA interface via a Web browser. The service user intends to define and execute a workflow presented in Figure 2 for a set of 500 molecules, which is retrieved from MDL Available Chemicals database.[16] The MDL database query

can be constructed using Marvin Sketch[17] within the SOMA interface, or the molecules can be uploaded, e.g., as an SDF-file or a multi-mol2 file. The exemplary workflow consisting of four executing nodes calculates molecular properties and the probability of the molecules to bind to the target protein.

Execution node 1 converts the 2D molecular structures to 3D using CORINA,[18] producing, e.g., a maximum of three 3D conformations per molecule. Execution node 2 docks the 3D structures to the target protein using GOLD[19] with standard parameters, producing a maximum of 10 docked poses for each conformation generated in STEP 1. Execution node 3 rescores all docked molecular structures using XSCORE.[20] Execution node 4 calculates water solubility of the molecules using VOLSURF.[21]

After submission of the workflow, SOMA executes CORINA on a SUN/SOLARIS server, and the input molecules are converted to maximum of 1500 three-dimensional molecular structures depicting conformations within a defined energy window. SOMA then proceeds to dock this set with GOLD through a batch queue on an IBM/AIX server and ends up with a maximum of 15 000 docked conformations. To help finding representative ligand−protein complexes, XSCORE is run on a LINUX cluster using the docked conformations as input.

The number of molecules that are input to VOLSURF depends on the execution order of the programs in the workflow. In terms of computational efficiency, it does not make sense to pass 15 000 docked structures to VOLSURF to predict their water solubility, since the prediction is independent of the 3D conformation of the molecules. In this respect, SOMA can help to optimize the computational workflow. VOLSURF can be executed as an independent execution node in the SOMA workflow (Figure 2), and the program results can be later associated with the molecular structures to which they are applicable. For example, a value from a single water solubility calculation can be inherited to 10 docking solutions of the same molecular structure. Thus, the workflow has to do 500 VOLSURF calculations and still has a complete list of calculated molecular properties for all 15 000 molecular structures resulting from molecular docking.

The calculated data from the SOMA workflow is currently placed in one large CML (ascii)-file stored on a shared disk space in the CSC computing environment. The data could also be stored to a database, but this feature has not yet been implemented. The SOMA interface presents the results from the workflow as a table, where molecules are in the rows and the calculated values in the columns. Interface offers basic sorting and filtering of the results and the possibility of visualizing molecular structures.

Components. The four main components of the SOMA workflow (Figure 2) can be described as the following: (1) the program-XML eXtensible Markup Language (XML, see http://www.w3.org/XML) description for attaching scientific programs to SOMA and templates of the configuration files of the programs and batch queue systems; (2) the program *Grape*, which manages the workflow. In Grape, the results from one program (see the execution node, Figure 3.) can be used as input for the others; (3) the Toolkit that includes utility programs, e.g., to translate molecular structures to and from the system's internal XML data format, build the execution files for applications, and convert the calculated

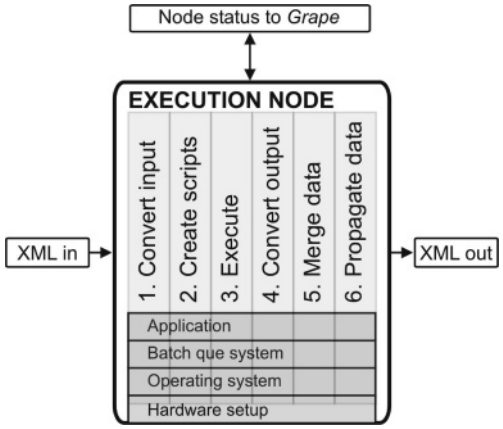

Figure 3. The main steps of the execution node. *Grape* moves the molecular data between the execution nodes in the XML format, which contains both the system information and the molecular information. Each execution node performs actions that are dependent on the hardware, the operating system, the batch queue system, and the application. The execution node 1.) converts the molecular structures from the CML format to the application input format and 2.) creates the scripts for running the application in the operating system, possibly utilizing the batch queue system. The program is then 3.) executed, and as soon as it finishes the application output is 4.) converted to the CML (structures) and XML (application results and metadata) formats using the SOMA toolkit. This CML and XML data are then 5.) merged together and subsequently merged to the input data that entered the execution node. Finally, special attention is paid to correctly 6.) propagate the computational data that can be propagated (see text) to the right molecules in the result repository (CML file or database) that collects the information from all of the execution nodes in the workflow.

results to CML (Chemical Markup Language) format; and (4) the Interface for user authentication, building the program interfaces and workflows, and visualization of the results.

The Program-XML. The SOMA Program-XML enables high level control of the required user input and description of the dependencies between the program control parameters. It is also designed to provide an easy way to create a Web browser interface for the scientific applications. For example, each parameter has an attribute that defines whether the parameter will be asked from every user or just from the advanced user. This feature enables tuning of the automatically generated Web browser interface of the application for users with different levels of expertise.

The program-XML also describes the service and run information of the program. The service information defines (a) the requirements of the input data before the program can be started and (b) the features of the program output data. For example, a docking program requires that the input molecules are in "3D coordinates". A feature of the 2D−3D conversion program output in the program-XML is "3D coordinates". With this knowledge, SOMA can inform the workflow management program *Grape* that output from the conversion program can be used as input for programs that require "3D coordinates". A program can have multiple requirements and features. This is a useful quality for checking the sequence in which programs can be run in the workflow. For example, starting to dock the 2D-structures of ligands to a protein is certain to produce erroneous results. These types of false workflows can be eliminated with the program-XML requirement descriptions. If the molecule trying to enter an execution node does not have 3D

coordinates, the SOMA workflow can realize this and inform the user that a program like CORINA[18] (generates 3D-structures) should be included with the workflow before proceeding to docking.

Despite the versatile use of the program-XML, it is relative easy to write, since it does not require programming skills. We have included the following programs and functionalities in the SOMA workflow: CORINA[18] to generate 3D coordinates, ROTATE[22] to create molecule conformations, BRUTUS[23] for superposition of ligands based on molecular interaction fields, GOLD[19] for docking, SYBYL[24] tools to calculate 2D and 3D properties of molecules, VOLSURF[21] to predict ADME(T) properties of molecules, and X-SCORE[20] for rescoring of docked ligand poses. The SOMA program-XML schema is provided as Supporting Information.

***Grape*: Execution of a Workflow.** The program *Grape* is the part of the SOMA workflow that takes care of the workflow execution and checks the integrity of the workflow (Figure 2).

The Web browser interface is used to define the workflow. The user is asked to provide parameters and input data needed by the programs in the workflow, as described in the program-XML. Molecules can be provided from the users local resources, drawn, or queried from the databases attached to SOMA. After providing the required input, the user will not need to interact with the system until the workflow is completed.

Before the workflow is submitted, the validity of the input molecules is checked, and they are converted to the CML format. The scientific applications are joined to the workflow in *Grape* as modules called execution nodes, which are then executed in logical order. Each execution node reports its status to *Grape* and submits the results to a data repository (shared disk or database) in the CML format. *Grape* will carry out the data transfer between the execution nodes in the internal XML format that contains the system specific information and the molecule structures and data in the CML format. *Grape* keeps track of the progress of the nodes and the entire workflow. *Grape* can use the data that accumulate from the execution nodes in the CML format as input for the remaining execution nodes.

The Execution Nodes in *Grape*. The execution of one particular application in the workflow is performed in the execution node. This is the level where the input and output of different applications are matched using the utility programs provided by the SOMA toolkit. The execution node thus does much more than just run the specific application.

The small molecules submitted to the SOMA workflow are, outside the execution node, always transferred in the XML (CML) format. The execution nodes can require user input, e.g., protein structures used for docking, which can be transferred to the node in any format. File format conversions must be done in the beginning and in the end of each execution node because third-party programs often require and produce inflexible application specific file formats for presenting molecular structures and properties. After the required format conversions, the execution node creates scripts for running the application in the operating system and writes the required application configuration files. This is a nontrivial step for drug discovery applications since the number and the names of molecules entering the execution node are not necessarily known before the previous execution node in the workflow has been executed. For example, the number of molecule structures produced by the execution node that generates molecular conformations is unknown before the execution node has finished its task.

After the execution node has executed the program, the resulting molecule structures will be converted back to the CML format. At that stage the information that identifies the execution node that produced the structures is attached to the names of the molecules. This way the workflow can keep track of the operations that SOMA does to the initial structures that enter the workflow.

It is possible to define ancestor and child structures for the molecules passing through the SOMA workflow. SOMA propagates data produced in the execution nodes according to this information. For example, the properties calculated using the 2D structure of the molecule are valid for all molecules having the same (ancestral) structure. SOMA can propagate data for both ancestors and for children, depending on which stage of the workflow the property is calculated. To apply the rules for the propagation of data, the results from each program have to be first written in the XML format (results-XML). SOMA also adds metadata like the protein structure used in docking to the results-XML. The results-XML will be merged with the CML data containing the molecule structures that are produced in the same execution node. Finally, the newly formed CML includes the molecular structures and the calculated results. This CML data will be merged with the data that entered the execution node (containing ancestor molecule structures with the merged results from the previous execution nodes). This way SOMA proceeds through every execution node in the workflow until the entire execution graph is complete.

The Interface. SOMA is used via a Web browser interface. The interface provides an access to the programs connected to the workflow and helps in the analysis and visualization of the results. The SOMA workflows are defined within the interface. After the workflow is complete, gathered results can be analyzed and postprocessed in the interface.

The interface to a program is automatically constructed from the information given in the program-XML description. It contains all the information needed to generate the Web browser interface for the application and get the required input from the user.

Attaching New Applications. To attach a new application to the SOMA workflow the service provider must produce the following information: (1) the program-XML for the application; (2) the template files for configuring the application (parsed using Template Toolkit (http://template-toolkit.org/); (3) the scripts for executing the application in UNIX/LINUX [The script uses the utility programs provided by the SOMA Toolkit to produce the configuration files, launches the application, and reports the status of the execution node to *Grape*.]; (4) the script to produce results-XML with the tools that are provided in the SOMA toolkit [Due to the high specificity of software output some tweaking is often required.]; and (5) the postprocessing script, which performs the steps 4−6 in Figure 3. The script mainly calls utility programs provided in the SOMA toolkit. The addition of a typical program, for example docking program GOLD,

to the SOMA workflow required approximately 2 days of work.

Technical Implementation. The SOMA Web browser interface and *Grape* can be run on the same server (e.g., RedHat 9). Currently the user authentication is done using CSC's extranet service called the Scientist's Interface (http://www.csc.fi/proj/sui/index.phtml.en). The SOMA interface is constructed using Perl. *Grape* requires Java version 1.5.

The scientific applications that are part of the SOMA workflow can be installed on any UNIX/LINUX-environment, even on a Grid. The only requirement is a secure connection (currently an SSH connection without a password) from the server running *Grape* to the application server.

DISCUSSION

The key scientific aspect of computational experiments is the logic of combining different software and algorithms in a sensible way to, e.g., predict the physicochemical properties of molecules or their binding probability to a protein receptor family. These are important and repeating processes in many molecular design projects. Through automation of technical routines involved in similar computational processes, SOMA offers significant advantages compared to traditional systems where the programs are installed as such and even offered on different server platforms.

The presented middleware brings benefits to both scientific service users and service providers. Users gain a better ability to link together programs from different vendors under one interface. Users can build computational workflows that characterize molecular properties using a selection of programs available at the scientific service provider. With the developed program description procedure, it is possible to attach a wide selection of software together. With the description, the SOMA workflow can extract and transfer molecular data between programs and presents and organizes the data in an extranet interface, thus automating many technical details of the underlying computing environment. The possibility to automate repeating steps in computing experiments can save a significant amount of time and gives the user the means to share the knowledge of useful workflows with her colleagues in a machine-readable format.

The SOMA workflow is useful for the service providers for the maintenance and building of versatile software environments. Once the proper use of the software in the computing environment has been deciphered by the application scientist, the knowledge is documented in a machine readable format and can be shared with the customers of the service. This enables first time users of software to have automated access to examples of how programs are configured and operated on a complex computing environment containing multiple UNIX platforms and batch queue systems. Examples also open great opportunities for teaching and can automate routine support work in molecular modeling.

The time invested in making machine-readable modeling program descriptions can have an impact on the service quality as well. The service provider can hide technological details of the system where the scientific services are run. For example, the maintenance of the service infrastructure is more transparent as program versions can be changed without notable interruptions to the service. The program descriptions are also important for maintaining legacy information: part of the knowledge remains with the scientific service provider even after the application scientist who configured the application has left.

Also advanced users can gain, for example, by sharing their knowledge of computational workflows to colleagues by configuring template workflows for a particular problem. Any automation cannot offer all the options that an advanced user wants to tune. However, the system is adjustable from the UNIX prompt, and users may reconfigure the automatically generated files before submitting the workflow for processing.

The main difference of the SOMA workflow compared with the existing molecular modeling workflows and data flows in chemistry and molecular design is the program description. Software can be integrated to the SOMA workflow by providing the software description. No programming skills are required to attach new software to SOMA. This makes the system more open for user modifications compared to commercial cheminformatics workflow solutions, e.g., Pipeline Pilot from Scitegic. The software attachment through program description also makes the difference from in-house workflow solutions developed in, e.g., pharmaceutical companies.[4,10] Rather than just offering ready-made workflows of cheminformatics programs, the SOMA workflow offers a set of tools that are used to create and customize workflows according to the third-party program selection available at the service provider. The descriptions of programs can be written by the person maintaining the program. The program-XML file and scripts describe the data that the program needs to operate and what it produces. The SOMA internal automation takes care of the data conversions with extra care that the conversion is correct. The program-XML description gives much flexibility for the scientific service provider over the platforms where the calculations are actually executed. For example, in the program description for molecular docking software it is possible to choose (and even dynamically change) the platform where the program is executed, e.g., UNIX/LINUX server, a computing Grid or Web service that can communicate with programmatic means.

CONCLUSION

We have created a workflow of small molecule calculations on a multiplatform computing Grid that improves the efficiency of using computational tools in designing molecular structures with desired properties. Choosing the right scientific software for the workflow is its own challenge, as modeling complex molecular systems requires multidisciplinary knowledge of biology, chemistry, and physics. Together with the described information technology challenges, building an effective environment requires a team of specialists, each with complementary expertise. An effective environment should have a maintainable software and hardware infrastructure that enables the specialists to share their specific skills and knowledge to the other members of the team through a common framework. If all human, computer, and software resources cooperate, described workflows potentially produce massive data sets, which must be turned into knowledge. This is our future objective: when the molecular structure and property

calculation workflows are technically easier to accomplish, the challenge switches more into the data management and data visualization problem—how one can make decisions based on the computed data.

The SOMA workflow is currently in production use for the academic research community in Finland. For source code enquiries, please contact the corresponding authors directly. Further information can be found from the project homepage http://www.csc.fi/proj/drug2000.

ACKNOWLEDGMENT

This work was funded by the Finnish Technology Agency TEKES DRUG2000 program grants and the Finnish IT Center for Science CSC. The SOMA programming team Jakub Järvenpää, Tapani Kinnunen, Antti Niemelä, Matti Parviainen, and Antti Virolainen are acknowledged for their contributions to this work. Dr. Olli Pentikäinen has contributed constructive discussion and criticism.

Supporting Information Available: The SOMA program-XML schema. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Garcia Castro, A.; Thoraval, S.; Garcia, L.; Ragan, M. Workflows in bioinformatics: meta-analysis and prototype implementation of a workflow generator. *BMC Bioinformatics* **2005**, *6* (1), 87.

(2) Gobbi, A.; Funeriu, S.; Ioannou, J.; Wang, J.; Lee, M.-L.; Palmer, C.; Bamford, B.; Hewitt, R. Process-driven information management system at a biotech company: concept and implementation. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (3), 964−975.

(3) Oprea, T. I.; Gottfries, J.; Sherbukhin, V.; Svensson, P.; Kuhler, T. C. Chemical information management in drug discovery: Optimizing the computational and combinatorial chemistry interfaces. *J. Mol. Graphics Modell.* **2000**, *18* (4), 512−524.

(4) Rojnuckarin, A.; Gschwend, D. A.; Rotstein, S. H.; Hartsough, D. S. ArQiologist: An Integrated Decision Support Tool for Lead Optimization. *J. Chem. Inf. Model.* **2005**, *45* (1), 2−9.

(5) Triballeau, N.; Acher, F.; Brabet, I.; Pin, J.-P.; Bertrand, H.-O. Virtual screening workflow development guided by the "receiver operating characteristic" curve approach. *J. Med. Chem.* **2005**, *48* (7), 2534−2547.

(6) Higginson, P. D.; Sach, N. W. High-Throughput Experimentation in Pharmaceutical Process R&D: Developing a New Software Workflow to Overcome Downstream Data-Analysis Bottlenecks and Improve Productivity. *Org. Process Res. Dev.* **2004**, *8*, 1009−1014.

(7) Stevenson, J. M.; Mulready, P. D. Pipeline Pilot 2.1. Computer Software Review. *J. Am. Chem. Soc.* **2003**, *125*, 1437−1438.

(8) Söderholm, A. A.; Lehtovuori, P. T.; Nyrönen, T. H. Three-dimensional structure−activity relationships of nonsteroidal ligands in complex with androgen receptor ligand-binding domain. *J. Med. Chem.* **2005**, *48* (4), 917−925.

(9) Järvenpää, J. A general workflow framework and its application to computational molecular modeling. Master's Thesis, Helsinki University of Technology, Espoo, 2005.

(10) Watson, P.; Verdonk, M.; Hartshorn, M. J. A web-based platform for virtual screening. *J. Mol. Graphics Modell.* **2003**, *22*, 71−82.

(11) Project Open Babel 1.100.2. http://openbabel.sourceforge.net/.

(12) Murray-Rust, P.; Rzepa, H. S. Chemical Markup, XML, and the World Wide Web. 4. CML Schema. *J. Chem. Inf. Model.* **2003**, *43* (3), 757−772.

(13) Gouet, P.; Emmanuel, C. ENDscript: a workflow to display sequence and structure information. *Bioinformatics* **2002**, *18* (5), 767−768.

(14) Project Taverna 1.2. http://taverna.sourceforge.net/.

(15) Stevens, R. D.; Robinson, A. J.; Goble, C. A. myGrid: personalised bioinformatics on the information grid. *Bioinformatics* **2003**, *19* (Suppl.), i302−i304.

(16) Program MDL ISIS Discovery Framework. http://www.mdl.com/products/framework/isis/.

(17) Program Marvin Sketch. http://www.chemaxon.com.

(18) Program CORINA 3.2. http://www.mol-net.de/software/corina/.

(19) Program GOLD 3.0. http://www.ccdc.cam.ac.uk/products/life_sciences/gold/.

(20) Program X-SCORE 1.2. http://sw16.im.med.umich.edu/software/xtool/manual/intro.html.

(21) Program VOLSURF 4.1. http://www.moldiscovery.com/soft_volsurf.php.

(22) Program ROTATE. http://www.mol-net.de/software/rotate/.

(23) Tervo, A. J.; Rönkkö, T.; Nyrönen, T. H.; Poso, A. BRUTUS: optimization of a grid-based similarity function for rigid-body molecular superposition. 1. Alignment and virtual screening applications. *J. Med. Chem.* **2005,** *48*, (12), 4076−4086.

(24) Program Tripos SYBYL 7.0. http://www.tripos.com/.

CI050388N