3rd International Workshop on Science Gateways for Life Sciences (IWSG 2011), 8-10 JUNE 2011

A Pipeline Pilot based SOAP implementation of FlexScreen for High-Throughput Virtual Screening

Horacio Pérez-Sánchez[1], Ivan Kondov[2], José M. García[1], Konstantin Klenin[2]and Wolfgang Wenzel[3,*]

[1]Computer Engineering and Technology Department, University of Murcia. Murcia (Spain)

[2]Steinbuch Centre for Computing, Karlsruhe Institute of Technology. Karlsruhe (Germany)

[3]Institute of Nanotechnology, Karlsruhe Institute of Technology. Karlsruhe (Germany)

ABSTRACT

Methods for in-silico screening of large databases of molecules increasingly complement and replace experimental techniques to discover novel compounds to combat diseases. As these techniques become more complex and computationally costly we are faced with an increasing problem to provide a community of life-science researchers with a convenient way to run complex high-throughput virtual screening (HTVS) calculations on distributed computing resources. To this end, we recently integrated the biophysics based drug screening methodology FlexScreen into a service applicable for large-scale parallel screening and reusable in the context of scientific workflows. Our implementation, based on Pipeline Pilot and SOAP provides an easy-to-use graphical user interface to construct complex workflows which are executed on distributed computing resources, thus accelerating the throughput by several orders of magnitude.

1 INTRODUCTION

The discovery of new drugs can be drastically accelerated with the use of high-throughput virtual screening (HTVS) methods (Friesner, et al., 2004; Halgren, et al., 2004; Meng, et al., 1992; Merlitz, et al., 2003; Merlitz and Wenzel, 2002; Merlitz and Wenzel, 2004) ongoing trend in medical research taking advantage of recent advances introduced in the field. In order to identify promising candidates for new drugs, chemical compound databases with millions of ligands (Irwin and Shoichet, 2005) need to be screened using HTVS against structurally resolved receptors and hence the access to computational resources becomes a serious issue. Many research organizations have access to high performance computing (HPC) resources distributed in computing grids and clusters, which can tremendously help to overcome these constraints (Perez-Sanchez and Wenzel, 2011).

HPC resources consist in a wide range of hardware and software resources for the research group members. They are usually accessed through well-defined gateways, which are based on web services or remote-access user interface machines (UIs). However, both solutions still require in-depth knowledge in grid technologies from the non-expert end users. The major drawback of this direct approach of doing scientific research is related to its complexity and difficulty of use making the learning curve too steep. Many efforts have then to be made to hide the complexity embedded in "the Grid" and to provide high-level services that allow scientists to take more effectively further advantage of the distributed resources.

Science gateways are the primary solutions dedicated to bridge such knowledge gaps. A Science gateway is defined as "a community developed set of tools, applications, and data that is integrated via a portal or a suite of applications, usually in a graphical user interface, that is further customized to meet the needs of a targeted community" (Catlett, 2002; Catlett, 2005). With science gateways non-grid-aware users can use grid infrastructure to run shared, well-tested applications customized for their own research field. Generally these solutions contain a set of research-specific applications developed by (and for) the community, and provide services integrated in a unified user interface, usually a web portal or a stand-alone graphical user interface. In the context of HTVS this problem is paramount because the target user community consists of pharmacists and biologists not trained or experienced in the use of HPC/grid infrastructures.

Very often, science gateways provide special higher-level services for construction and execution of scientific workflows, i.e., means to automate processing of multiple steps in parallel or in a sequence, including branching and loops. Thus, workflows are abstract logical maps of the complex simulation protocols. Scientific workflows require each step (often a different scientific application) to provide common interfaces for execution and data exchange. Currently, several systems for workflow management are employed in different projects. For example, the UNICORE workflow engine has been used in the area of QSAR/QSPR (Sild et al. 2005), Gridbus for brain imaging (Pandey et al. 2009). Other very widely used workflow systems are Kepler (kepler-project.org) and Taverna (taverna.org.uk). For a review on scientific workflows we refer to (Yu et al. 2005).

In order to make HTVS methods accessible for the relevant community, we must (a) integrate the screening method into an easy-to-use graphical interface (b) the interface must be reusable in different scientific workflows in combination with other applications and (c) provide a seamless access to large-scale computational resources to enable large screening campaigns. In this work we present a solution for the HTVS application FlexScreen which

*To whom correspondence should be addressed.

takes into account these three aspects. In Section 2 we introduce the program FlexScreen as well as the methods we employed to integrate FlexScreen into workflows for HTVS. In Section 3, we will particularly describe how we adopted Pipeline Pilot and the SOAP standard to implement our concept and present a case study with use of the developed machinery. In Section 4 we will conclude and give an outline of future work.

2 METHODS

2.1 FlexScreen

HTVS calculations have been performed with the all-atom receptor−ligand docking program FlexScreen (Guerrero, et al., 2011; Merlitz, et al., 2003; Merlitz and Wenzel, 2002), which employs a force-field based scoring function (similar to Autodock (Morris, et al., 1996)) and a Monte-Carlo based search algorithm based on the stochastic tunneling method (Wenzel and Hamacher, 1999), which has the advantage that it suffers only a comparatively small loss of efficiency when an increasing number of receptor degrees of freedom is considered.

A physical model is implemented which takes implicitly into account the influence of the solvent in the interaction between ligands and proteins. The free energy of the system includes vacuum contribution that has been previously available in FlexScreen as well as additional solvation terms for the individual species and for the complex as a linear sum of atomic parameters (Eisenberg, et al., 1984). This latter model has the advantage that it is faster than other methods presently used and still has proven to be reasonably accurate. The solvent accessible surface area of the molecules must be determined, which is a computationally intensive task, and in this work an exact and an approximated, but less time consuming approach are presented. The other main contribution of this approach is the determination of the weight parameters for very different atom and bond types, being them derived from experimental partition coefficients data in the cases octanol−water and gas−water.

2.2 Pipeline Pilot

Pipeline Pilot (http://www.scitegic.com) provides services and a workflow engine basing on Service Oriented Architecture (SOA) (Yang, et al., 2010) allowing very effective workflow life-cycle management, i.e. it ensures maximum reuse of already integrated modules. In addition, it supports SOAP with Web Services Description Language (WSDL) extensions for efficient decoupling of workflow management from services' internal implementation. In this way, in addition to its built-in functionality, the architecture of Pipeline Pilot has been organized for integration and extensibility and designed to interoperate with external software objects and applications. A number of mechanisms are available to automate the execution of a remote program. Additional options are available if the screening code resides on the workflow server. In general, two mechanisms are used for remote execution. Simple integrations use Telnet and File Transfer Protocol (FTP). More complex integrations use Simple Object Access Protocol (SOAP)(Snell, et al., 2002) and web services. SOAP provides a way for applications to communicate with each other over the HPC resources. The SOAP framework is independent of any particular programming model, environment, or language. It is a structured method for sharing messages between server and client, and relies on XML to define the format of the information and then adds the necessary HTTP headers to the information. Most applications do not deal directly with the underlying SOAP data structures. Instead, they use a toolkit specific to their programming language and operating system. The toolkit simplifies the process of making SOAP calls and processing the returned results.

Pipeline Pilot provides several integration methods so that several applications existing either in the workflow server, remote server or cluster can be executed automatically in a workflow. Pipeline Pilot provides also data integration tools that assist in the assembly of information from different formats and pertaining to different databases. A convenient and intuitive graphical user interface via a web browser is provided for constructing and executing the workflows. The workflows are assembled using modules that are represented as icons in the graphical user interface. The workflows are actually stored in an XML format and can be easily exchanged between users. The modules, called components, include a variety of data readers, manipulators, calculators, data viewers, and data writers. For example, there are convenient data reading modules for ISIS files, SD-files, and SMILES, as well as delimited text and Excel spreadsheet files. Data viewers and writers include standard applications, such as WebLabViewerPro and Spotfire. An HTML molecular table viewer provides a convenient way to view tabular results with chemical structures. Although the applicability of the pipelining provided by this software is generic, the numerous (>200) specific components provided by SciTegic are heavily geared toward chemoinformatics environments. For academic users there is a free version of Pipeline Pilot available.

2.3 Workflows and Data Pipelining

A workflow in Pipeline Pilot refers to the way a protocol is defined, usually in form of several disconnected pipelines, each of which is made of components joined by pipes. A component refers to an individual operation to be performed on a set of data records. The order of execution depends on the order the components are joined since the protocols are executed from left to right, top to bottom.

In the specific form of a workflow called data pipelining, records are passed individually down the pipes. Data pipelining allows the automation of the HTVS process and the integration of several related modeling and database packages. Thus, in addition to orchestration of multiple workflow steps the data pipelining provides means for seamless data exchange between the individual application modules. The end users' work in HTVS projects can be enormously facilitated by the exploitation of already prepared sets of commonly used collections of tasks in the form of workflows. These protocols can be later deployed on HPC resources in a simple and automated fashion. An advantage of the pipelining approach is the ability to capture and conveniently share workflows for better reuse.

3 IMPLEMENTATION AND USE CASES

3.1 Pipeline Pilot Modules for FlexScreen

FlexScreen was initially designed as a standalone command line application. In the first part of the work reported here we have implemented a set of Pipeline Pilot modules that are required to run FlexScreen within Pipeline Pilot. The required executables and template configuration files are placed in the Pipeline Pilot server.
The FlexScreen integration in Pipeline Pilot is depicted in Fig. 1. In pipelines 1 and 2 end users need to specify receptor and ligand database files in the molecular standard PDB format. If the user works with other molecular formats (smi, sdf, etc.), the protocol can be easily modified using molecular format converters included in the standard components collection of Pipeline Pilot. Afterwards the initial receptor and ligand files can be parameterized depending on the charge model used, hydrogen model, etc. and additional components (pH, tautomers, etc.) can also be easily included in the pipeline. Once the molecules are ready for the HTVS calculations, the docking parameters (degree of flexibility, simulation length, physical model, etc.) and parallel calculation parameters (batch

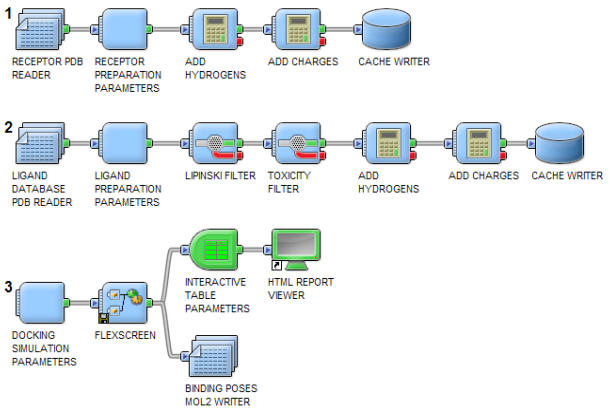


Figure 1: Integration of FlexScreen into Pipeline Pilot workflows. Pipelines 1 and 2 read and format the ligand database and receptor files. In Pipeline 3 the input molecules are received and the docking simulation parameters are specified. Then the FlexScreen component performs the SOAP calls and runs the calculations on the HPC resources. Finally the results are processed and presented in an interactive table format.

DOCKING RESULTS OF streptavidin

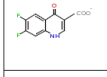
Molecule	Name	SourceTag	DOCK_STATE	INI_ENERGY	ATOMS	BONDS	FINAL_ENERGY_1	FINAL_ENERGY_2	RMSD
	jnk.vs.1-81_1	results.mol2 results.info	1	0.000000	20	0	-88.777329	-88.777329	1.776939
	jnk.vs.1-82_1	results.mol2 results.info	1	6.246283	42	1	-58.177132	-58.389315	2.002606
	jnk.vs.1-83_1	results.mol2 results.info	1	0.000000	33	0	-16.738663	-16.738663	1.990425
	jnk.vs.1-84_1	results.mol2 results.info	1	0.000000	31	0	-49.256140	-49.256140	1.788244

Figure 2: Sample of the output results in HTML format, directly from the web browser. HTVS results are presented in consecutive rows for the different ligands of the database. Different columns contain information about each ligand regarding name, energy calculations, RMSD, etc. Clicking on each ligand 2D representation opens a new window with detailed information about the 3D ligand binding mode as shown in Figure 3.

size, number of processors to use, etc.) are specified at the beginning of the third pipeline. In any case the protocol provides default parameters for all the components so that the user only needs to select ligand, receptor and binding site parameters.

One of the challenges in a virtual screening experiment is to analyze and organize the returned results. Again, an expert modeler will be familiar with tools available within a modeling environment to examine and filter the results. For an end user, the analysis and presentation must be automated so that they can correctly generate the information that they need for further decision making. Using a single PC as a server, a single user is thus able to design and run application workflows that link all available Pipeline Pilot modules with FlexScreen for HTVS.

3.2 SOAP Implementation of FlexScreen

The integration in Pipeline Pilot alone is, however, insufficient for really large in-silico screening campaigns. The improved accuracy of FlexScreen comes at the price of the computation cost of the underlying biophysical model. Therefore, we have implemented the FlexScreen Pipeline Pilot modules as a SOAP-based (Snell, et al., 2002) service capable to run on large distributed architectures, such as computing grids and clouds. We have developed SOAP-based web services for the remote FlexScreen application using software such as Apache / Tomcat (http://tomcat.apache.org) or the Perl SOAP::Lite module (http://soaplite.com). The SOAP wrapper contains sufficient processing functionality to perform the following tasks:

1. Receive a batch of ligands and receptor file as a SOAP message and save them to a file. One of the advantages of using SOAP is that it allows a batch size to be specified, allowing the collation a series of individual docking requests in a single request for efficiency.
2. Receive complementary information as SOAP messages and save it to files, e.g., protein active site, configuration files related to simulation parameters, etc.
3. Execute FlexScreen on the server and HPC resources using the files previously created.
4. Read the resulting files and pass them back as a SOAP message to the calling component. A report on the results will be automatically prepared as an interactive HTML report, a PDF document, or a spreadsheet.

3.3 Examples of Use

Results from a HTVS calculation performed by an end user are shown in Figs. 2 and 3. As seen in Fig. 2 the resulting data is clearly organized in tables which are directly opened in the web browser after the screening calculations. The user can control the degree of detail in the final report interacting with the “table parameters” component as well as reorganize easily and sort the final data with a few mouse clicks in the web browser. There is also the possibility of exporting the results to other standard formats, i.e., PDF, Word, Excel spreadsheets, CSV text files, etc.

From the perspective of users’ experience, we found that the access to well-developed and validated workflows using FlexScreen encourages the user to test and explore new ideas. Informal discussions with users who have performed HTVS calculations with FlexScreen in this way confirms that the deployment of HTVS methods does not just get the same answers faster, but that scientists end up asking many more “what-if” questions and running many more experiments than they would have done when a modeler had to be involved in each case.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we have described the implementation of a HTVS methodology in a science gateway environment making use of the workflow environment provided by Pipeline Pilot. The solution basing on SOAP and web services enables the exploitation of distributed HPC resources (grid computing). The only drawback of Pipeline Pilot is its commercial license for non-academic users. Now we are exploring several open source alternatives.

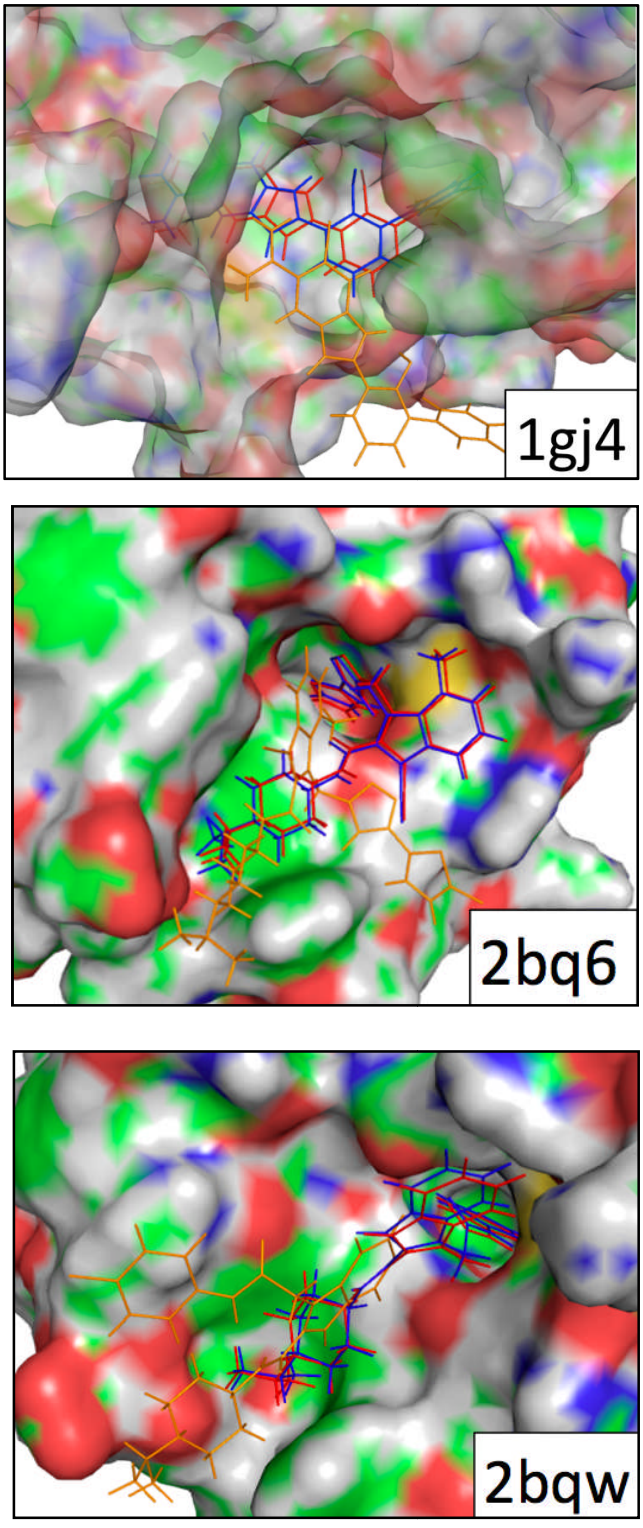


Figure 3: 3D representation of the HTVS results obtained for two different receptor-ligand pairs (PDB IDs 1gj4, 2bq6 and 2bqw). Blue color denotes the experimental ligand binding mode, orange color the FlexScreen prediction without considering solvation and the red color the prediction with the consideration of solvation.

ACKNOWLEDGEMENTS

This research was supported by a Marie Curie Intra European Fellowship within the 7th European Community Framework Programme (FP7 IEF INSILICODRUGDISCOVER), the FundaciónSéneca (Agencia Regional de Ciencia y Tecnología, Región de Murcia) under grants 00001/CS/2007 and 15290/PI/2010 and a postdoctoral contract from the University of Murcia (30th December 2010 resolution). I. K. acknowledges gratefully continuous support and funding by Programme "Supercomputing" of the Helmholtz Association.

REFERENCES

Catlett, C. (2002) The philosophy of TeraGrid: Building an open, extensible, distributed TeraScale facility, Ccgrid 2002: 2nd Ieee/Acm International Symposium on Cluster Computing and the Grid, Proceedings, 479, 8-8.

Catlett, C.E. (2005) TeraGrid: A foundation for US cyberinfrastructure, Network and Parallel Computing, Proceedings, 3779, 1-1.

Eisenberg, D., et al. (1984) Analysis of membrane and surface protein sequences with the hydrophobic moment plot, Journal of molecular biology, 179, 125-142.

Friesner, R.A., et al. (2004) Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy, Journal of medicinal chemistry, 47, 1739-1749.

Guerrero, G., et al. (2011) Effective Parallelization of Non-bonded Interactions Kernel for Virtual Screening on GPUs. In Rocha, M., et al. (eds), 5th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2011). Springer Berlin / Heidelberg, pp. 63-69.

Halgren, T.A., et al. (2004) Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening, Journal of medicinal chemistry, 47, 1750-1759.

Irwin, J.J. and Shoichet, B.K. (2005) ZINC--a free database of commercially available compounds for virtual screening, Journal of chemical information and modeling, 45, 177-182.

Meng, E.C., Shoichet, K. and Kunz, I.D. (1992) Automated Docking with Grid-Based Energy Evaluation, J.Comp.Chem., 13, 505.

Merlitz, H., Burghardt, B. and Wenzel, W. (2003) Stochastic tunneling method for high throughput database screening. Nanotech 2003, Vol 1.

Merlitz, H. and Wenzel, W. (2002) Comparison of stochastic optimization methods for receptor-ligand docking, Chemical Physics Letters, 362, 271-277.

Merlitz, H. and Wenzel, W. (2004) High throughput in-silico screening against flexible protein receptors. In Lagana, A., et al. (eds), Computational Science and Its Applications - Iccsa 2004, Pt 3. pp. 465-472.

Morris, G.M., et al. (1996) Distributed automated docking of flexible ligands to proteins: parallel applications of AutoDock 2.4, J Comput Aided Mol Des, 10, 293-304.

Pandey S, Voorsluys W, Rahman M, et al. A grid workflow environment for brain imaging analysis on distributed systems Chen J, Cafaro M, eds. Concurrency Computat.: Pract. Exper. 2009;21(16):2118-2139.

Perez-Sanchez, H. and Wenzel, W. (2011) Optimization Methods for Virtual Screening on Novel Computational Architectures, Current Computer-Aided Drug Design, 7, 44-52.

Sild S, Maran U, Romberg M, Schuller B, Benfenati E. OpenMolGRID: Using Automated Workflows in GRID Computing Environment. In: Sloot P, Hoekstra A, Priol T, Reinefeld A, Bubak M, eds. Advances in grid computing -- EGC 2005.Vol 3470. Springer Berlin / Heidelberg; 2005:464-473.

Snell, J., Tidwell, D. and Kulchenko, P. (2002) Programming Web services with SOAP. O'Reilly & Associates, Sebastopol, CA.

Wenzel, W. and Hamacher, K. (1999) Stochastic tunneling approach for global minimization of complex potential energy landscapes, Physical review letters, 82, 3003-3007.

Yang, X.Y., Bruin, R.P. and Dove, M.T. (2010) Developing an End-to-End Scientific Workflow. A Case Study Using a Comprehensive Workflow Platform in e-Science, Computing in Science & Engineering, 12, 52-61.

Yu J, Buyya R. A taxonomy of scientific workflow systems for grid computing. SIGMOD Rec. 2005;34:44-49.