

# Developing science gateways for drug discovery in a grid environment

Horacio Pérez-Sánchez, Vahid Rezaei, Vitaliy Mezhuyev, Duhu Man, Jorge Peña-García, Helena den-Haan, Sandra Gesing

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 the research community of life-sciences with a convenient tool for high-throughput virtual screening (HTVS) on distributed computing resources. To this end, we recently integrated the biophysics-based drug screening program 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 Simple Object Access Protocol (SOAP) provides an easy-to-use graphical user interface to construct complex workflows which can be executed on distributed computing resources, thus accelerating the throughput by several orders of magnitude.

# Developing Science Gateways for Drug Discovery in a Grid Environment

Horacio Pérez-Sánchez<sup>1</sup>, Vahid Rezaei<sup>2</sup>, Vitaliy Mezhujev<sup>3</sup>, Duhu Man<sup>4</sup>, Jorge Peña García<sup>1</sup>, Helena den-Haan<sup>1</sup>, and Sandra Gesing<sup>5</sup>

<sup>1</sup>Bioinformatics and High Performance Computing Research Group (BIO-HPC), Computer Engineering Department, Universidad Católica San Antonio de Murcia (UCAM), Murcia, Spain

<sup>2</sup>University of Economic Sciences, Institute for Research in Fundamental Science (IPM), Tehran, Iran

<sup>3</sup>Department of Informatics and Software Engineering of Berdyansk State Pedagogical University, Ukraine

<sup>4</sup>Dept. of Information Engineering, Graduate School of Engineering, Hiroshima University, Japan

<sup>5</sup>Center for Research Computing, University of Notre Dame, IN, USA

## 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 the research community of life-sciences with a convenient tool for high-throughput virtual screening (HTVS) on distributed computing resources. To this end, we recently integrated the biophysics-based drug screening program 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 Simple Object Access Protocol (SOAP) provides an easy-to-use graphical user interface to construct complex workflows which can be executed on distributed computing resources, thus accelerating the throughput by several orders of magnitude.

Keywords: Virtual Screening, FlexScreen, Science gateways, Drug Discovery, High Performance Computing

## 1 INTRODUCTION

2 Drug discovery can be drastically accelerated with the use of high-throughput virtual screening (HTVS)  
3 methods (Friesner et al., 2004; Halgren et al., 2004; Meng et al., 1992; Merlitz and Wenzel, 2002, 2004),  
4 an ongoing trend in medical research taking advantage of recent developments in algorithms and computer  
5 technology. In order to identify promising candidates for novel drugs, chemical compound databases  
6 with millions of ligands (Irwin and Shoichet, 2005) need to be screened using HTVS against structurally  
7 resolved receptors and thus distributing the workload on resources such as computing grids becomes  
8 essential. On the other hand, optimization of existing methods for HTVS to utilize novel high performance  
9 computer (HPC) architectures such as GPUs (Pérez-Sánchez and Wenzel, 2011; Sánchez-Linares et al.,  
10 2011b) can significantly reduce the run time per ligand.

11 Currently, HPC resources are mostly accessed remotely through low-level front-end machines (user  
12 interface machines) or using grid middleware and thus require from the non-expert end users in-depth  
13 knowledge of diverse batch systems or grid middleware protocols, respectively. To acquire the knowledge  
14 to use this complex low-level infrastructure for real-life applications makes the learning curve for scientists  
15 very steep. This is why efforts have still to be made to hide the complexity embedded in the Grid and to  
16 provide productive high-level services that allow scientists to take more effectively further advantage of  
17 the distributed resources.

18 Science gateways are the primary solutions dedicated to bridge such knowledge gaps. A science  
19 gateway is defined as *a community-developed set of tools, applications, and data that is integrated via*

20 a portal or a suite of applications, usually in a graphical user interface, that is further customized to  
21 meet the needs of a targeted community (Catlett, 2005, 2002). With science gateways non-grid-aware  
22 users can use grid infrastructure to run shared, well-tested applications customized for their own research  
23 field. Generally these solutions contain a set of research-specific applications developed by (and for)  
24 the community, and provide services integrated in a unified user interface, usually a web portal or a  
25 stand-alone graphical user interface. In the context of HTVS, this problem is paramount because the  
26 target user community consists of pharmacists and biologists not trained or experienced in the use of  
27 HPC/grid infrastructures.

28 Very often, science gateways provide special higher-level services for construction and execution of  
29 scientific workflows, i.e., means to automate processing of multiple steps in parallel or in a sequence,  
30 including branching and loops. Scientific workflows are abstract logical maps of complex simulation  
31 protocols and require that each step (often a different scientific application) provides common interfaces  
32 for execution and data exchange. Diverse mature science gateways or science gateway frameworks have  
33 evolved in different projects, which additionally allow for workflow management. For example, the  
34 UNICORE workflow engine and its workbench have been used in the area of Quantitative Structure-  
35 Activity Relationship (QSAR) and Quantitative Structure-Property Relationships (QSPR) models (Sild  
36 et al., 2005), and the Gridbus workflow for brain imaging (Pandey et al., 2009). Other very widely used  
37 workflow-enabled science gateways are Pipeline Pilot (<http://www.accelrys.com>), with different licensing  
38 options depending on the academic or industry version, Kepler (Ludäscher et al., 2006), Galaxy (Goecks  
39 et al., 2010; Blankenberg et al., 2010; Giardine et al., 2005), WS-PGRADE (Kacsuk et al., 2012),  
40 KNIME (Berthold et al., 2008) and Taverna (Wolstencroft et al., 2013) with open source licenses. For a  
41 review on scientific workflows we refer to Deelman et al. (2009).

42 To get an idea about the difficulties with the direct exploitation of HPC systems using HTVS methods  
43 we will describe how this process is usually carried out by expert users without use of science gateways.  
44 There are mainly three differentiated stages involved in the process:

- 45 1. Simulation data preparation: all the necessary data for the simulation must be conveniently prepared  
46 and the HPC system set up accordingly. In a classical parallel HPC system the total simulation is  
47 divided into different simulation units. Those units belong to thousands or more configuration files  
48 that must be arranged from a single file valid for the sequential execution of the program. This is  
49 not easy to do for end users, since it requires the use of different shell scripts for preparing those  
50 input files. Besides, specific configuration files for the queuing system must be set up for each  
51 independent simulation. Therefore, advanced knowledge of different IT technologies like tasks  
52 parallelization, input file structure, etc., is required at this stage.
- 53 2. Execution of the simulation: using different methods, the different simulation units are sent to  
54 the HPC system for their execution. The user needs to take care that there are no errors, to check  
55 continuously that the system is working properly and calculations are being performed seamlessly,  
56 and when the computations are finished, that there have been no errors.
- 57 3. Processing and interpretation of the results: it is usually necessary to move all the relevant data  
58 produced in the simulation to a local machine for its posterior analysis. Advanced knowledge of  
59 how HPC filesystems work is generally required at this stage. Lastly, all data needs to be curated  
60 and processed, normally using different advanced tools.

61 Given all the different and complex stages of the general simulation process, it is clear that for a user  
62 to be able to run calculations in this fashion, advanced knowledge of several tools, filesystems, etc., is  
63 required. Therefore, not all specialized users would be able to run computational experiments in this  
64 environment but only the advanced ones. This is why a work environment of another quality should be  
65 provided for the end user to exploit these resources effectively.

66 In order to make HTVS methods accessible for the relevant community, we identified the following  
67 goals. (i) The screening method has to be made accessible via an easy-to-use graphical interface; (ii)  
68 The HTVS application has to be integrated in such a way that it becomes reusable in different scientific  
69 workflows in combination with other applications; (iii) The screening method has to provide a seamless  
70 access to large-scale computing resources to enable large screening campaigns. In this work, we present a  
71 solution for the HTVS application FlexScreen which will take into account these three aspects. In general,  
72 our research belongs to the problem of subject adaptation of existing IT technologies, its customization

73 for known domain of expertise of end users. This is an expansion of our previous work (Pérez-Sánchez  
74 et al., 2011). In the next section we will consider requirements for development of such type solutions.  
75 In Methods Section, we will introduce the FlexScreen application as well as the methods we employ to  
76 integrate FlexScreen into workflows for HTVS. In Implementation Section, we will particularly describe  
77 how we adopted Pipeline Pilot and the Simple Object Access Protocol (SOAP) to implement our concept  
78 and present a study case with use of the developed machinery. Furthermore, we present our investigation  
79 of integrating the implemented methods in diverse workflow-enabled science gateways. In Conclusions  
80 and future work Section, we will conclude and give an outline of future work.

## 81 **PROBLEM STATEMENT**

82 Our research belongs to the general problem of subject adaptation of existing IT technologies, its  
83 customization for known domain of expertise of end users. For the moment there are several approaches,  
84 intended for simplification of using HTVS methods (the list of corresponding software tools can be found  
85 at Jacob et al. (2012)).

86 The idea of our research is to find the most effective way of customization of HTVS methods. Analysis  
87 of existing approaches allows us to formulate next preconditions for the possible solution:

- 88 1. Simple and easy development of results by users, having no specific IT knowledge and qualification;
- 89 2. Rapid development, as e.g. by following RAD (Rapid Applications Development) technology;
- 90 3. Possibility of quick redevelopment of a solution without changing IT infrastructure;
- 91 4. Flexibility of solution, i.e. it should not be hardcoded inside a corresponding tool and so allowing  
92 development of extensions by advanced users;
- 93 5. The solution should allow to a user easy and naturally express logic (properties and behavior) of a  
94 domain;
- 95 6. Generality of solution, i.e. possibility to take into account specifics of modeling different domains  
96 (as biology, chemistry, physics, geometry);
- 97 7. Possibility of sharing and reusing existing solutions;
- 98 8. The solution should allow feature analysis (best of all, by attracting visual techniques);
- 99 9. On-fly testing and verification of basic properties (incl. syntax check) before executing;
- 100 10. The implementation environment should be easy and commonly used.

101 Having these requirements we have analyzed existing technologies and corresponding software tools.

102 Simple development means that most of calculation issues (effectiveness, optimization, scheduling,  
103 resources etc.) are automated, and user can concentrate on the task in terms of his domain of expertise  
104 (e.g. domain specific data types). Details of the underlying programming code also should normally be  
105 hidden. At the same time solution should take into account different qualification of users, which can  
106 include biology scientists, persons with IT background and people from pharmacy companies. Thus  
107 solution should be flexible, and allow quick redevelopment to follow possible changes in HTVS methods.

108 To give to users the freedom in expression of logic we decided not to use point solutions - software  
109 tools having interface to HTVS, but introduce language, allowing to users to express logic of domain in  
110 the way they want to do it.

111 Here the problem of development of Domain Specific Languages (DSL) can be addressed. In general,  
112 the approaches for modeling domains can be divided into two parts: 1) using a so-called General Purpose  
113 Language (GPL) or 2) developing a DSL. Although existing GPLs are good for expressing computational  
114 domains, they are not suitable for modeling biological domains. At the same time, biological modeling  
115 approaches do not allow us to express data structures and computational processes.

116 Thus, we need to develop an approach, that allows us to express heterogeneous semantics of interlinked  
117 biological and computational domains. The main task here is to specify protocol of calculation, can be  
118 considered as workflow of tasks. In general, Workflow Management System (WMS) can be used here,

119 allowing to develop and manage different protocols as a sequence of tasks. Note, workflow approach is  
120 becoming more and more popular nowadays for modeling distributed IT environments, including grids  
121 and cloud computing. They allow to manage the execution of various distributed processes.

122 A scientific workflow system is a special type of a WMS, allowing to build protocols for some  
123 scientific application as simplified maps of complex simulation protocols. Development of scientific  
124 workflows for using HTVS methods will be considered as an effective solution for our research problem.

125 We choose Pipeline Pilot for implementation due to its flexibility, allowing to develop workflows for  
126 different domains. Pipeline Pilot can be defined as scientific visual and dataflow programming language,  
127 allowing construction and execution of scientific workflows. At the same time the Pipeline Pilot is simple  
128 enough to be used by people having no specific IT knowledge and skills.

129 Note, that using visual languages (VL) allows the manipulation of graphical objects as mathematical  
130 complexes or data structures. Using VL, e.g. visual programming languages, belongs to RAD technology.  
131 VLS are also effectively used for data analysis in quite different domains.

132 As most of VLS, Pipeline Pilot uses idea of drawing boxes and connecting them by arrows (pipes). It  
133 allows development in an interactive way and checking syntax on the fly. As for protocol of calculation,  
134 Pipeline Pilot implemented the idea of dataflow programming, emphasizing the movement of data through  
135 pipes. This approach allows users to automate parallelization.

136 Due to popularity, protocols developed in Pipeline Pilot enable scientists to publish scientific services  
137 making them available across scientific community. Moreover, Pipeline Pilot workflow language is a  
138 standard, which allows to encapsulate and deploy the best practices. So in general the proposed solution  
139 reduces not only development time, but also costs, spent by integrating with HTVS point solution software.

## 140 **METHODS**

### 141 **FlexScreen**

142 In this work, HTVS calculations have been performed with the all-atom receptor–ligand docking program  
143 FlexScreen (Merlitz and Wenzel, 2002; Kokh and Wenzel, 2008), which employs a force-field based  
144 scoring function (similar to Autodock (Morris et al., 1996)) and a Monte-Carlo based search algorithm  
145 based on the stochastic tunneling method (Wenzel and Hamacher, 1999). This method has the advantage  
146 that it suffers only a comparatively small loss of efficiency when an increasing number of degrees of  
147 freedom of the receptor is considered.

148 A physical model is implemented, which takes implicitly into account the influence of the solvent in  
149 the interaction between ligands and proteins. The free energy of the system includes a vacuum contribution  
150 that has been previously available in FlexScreen as well as additional solvation terms for the individual  
151 species and for the complex as a linear sum of atomic parameters (Eisenberg and McLachlan, 1986). This  
152 latter model has the advantage that it is faster than other methods presently used and has still proven to be  
153 reasonably accurate. The solvent accessible surface area of the molecules must be determined, which is  
154 a computationally intensive task. The other main advantage of the method is the determination of the  
155 weight parameters for different atom and bond types deriving from experimental partition coefficients in  
156 the cases of octanol–water and gas–water.

### 157 **Pipeline Pilot**

158 Pipeline Pilot provides for applications based on SOAP standard methods to communicate with each other  
159 over the HPC resources (Yang et al., 2010), allowing very effective workflow life-cycle management,  
160 i.e. it ensures maximum reuse of already integrated modules. In this way, in addition to its built-in  
161 functionality, the architecture of Pipeline Pilot has been organized for integration and extensibility and  
162 designed to interoperate with external software objects and applications. A number of mechanisms are  
163 available to automate the execution of a remote program. Additional options are available if the screening  
164 code resides on the workflow server.

165 Different mechanisms are used for remote execution ranging from simple Telnet and File Transfer  
166 Protocol (FTP) up to more elaborated standards such as SOAP (Snell et al., 2002) and web services. The  
167 SOAP standard provides methods for applications to communicate with each other over the HPC resources.  
168 The Pipeline Pilot supports SOAP with Web Services Description Language (WSDL) extensions for  
169 efficient decoupling of workflow management from the internal implementation of services. The SOAP  
170 framework is independent of any particular programming model, environment, or language. It is a  
171 structured method for sharing messages between server and client, and relies on the language XML

172 to store and transmit the information and adds the necessary HTTP headers to the information. Most  
173 applications do not deal directly with the underlying SOAP data structures. Instead, they use a toolkit  
174 specific to their programming language and operating system. The toolkit simplifies the process of making  
175 SOAP calls and processing the returned results.

176 RESTful services gain popularity, which typically work more faster comparatively with SOAP  
177 implementations. At the same time, it is more difficult to broadcast RESTful services, which is a  
178 significant point in the context of development of scientific gateways. SOAP provides an interface for  
179 WSDL, allowing to define complex protocols, which is exactly the case of using Pipeline Pilot. So REST  
180 and SOAP have their own advantages and drawbacks and both are intensively used in modern web-based  
181 systems. The decision to choose the needed protocol can depend on problem domain only.

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

## 196 **Workflows and Data Pipelining**

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

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

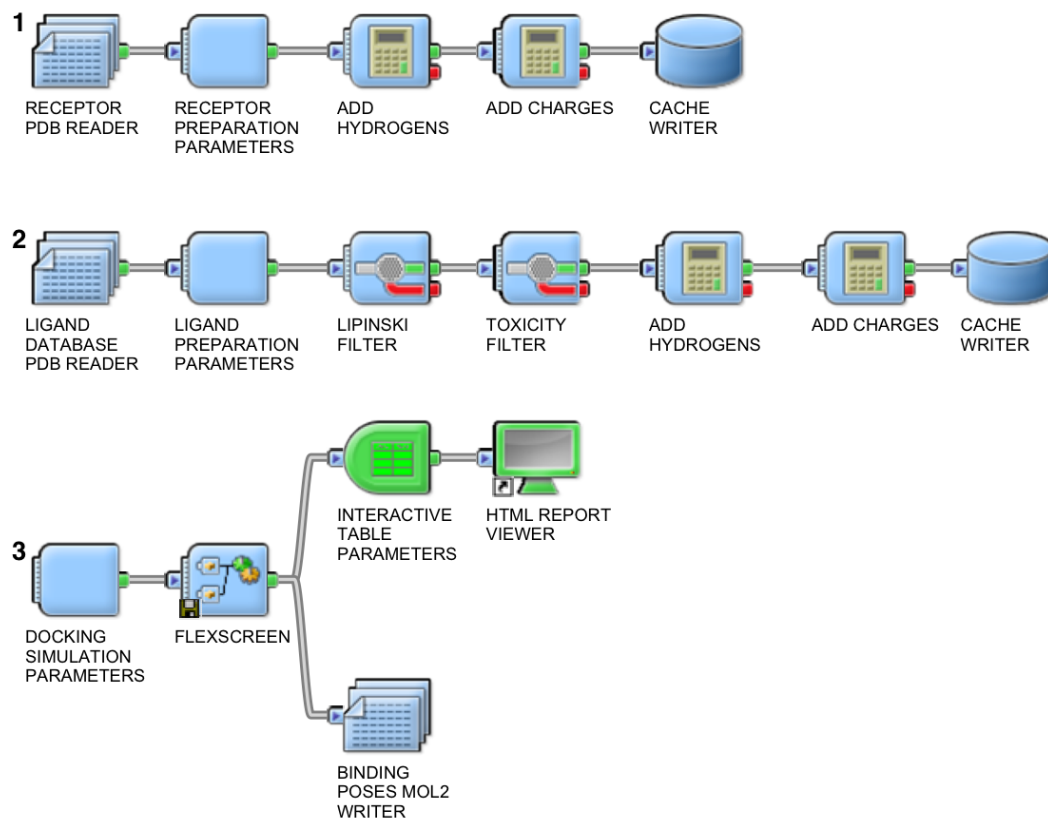
## 209 **IMPLEMENTATION**

### 210 **Pipeline Pilot Modules for FlexScreen**

211 FlexScreen was initially designed as a standalone command line application. Therefore, most previous  
212 users are familiar with this method of running it. The GUI that the gateway provides is meant for those  
213 users who are not familiar with running command line applications. Nevertheless, the gateway is meant  
214 for both kinds of users. From one side, users that were familiar with the command line version, will not  
215 find any difficulty when using the GUI, whereas new FlexScreen users or users that have never used any  
216 command line application will be quickly able to setup docking simulations.

217 In the first part of our work we implemented a set of Pipeline Pilot modules that were required to  
218 run FlexScreen within Pipeline Pilot. The required executables and template configuration files were  
219 placed in the Pipeline Pilot server. The FlexScreen integration in Pipeline Pilot is depicted in Fig. 1. In  
220 pipelines 1 and 2 end users need to specify receptor and ligand database files in the standard molecular  
221 PDB format. If the user works with other molecular formats (smi, sdf, etc.), the protocol can be easily  
222 modified using molecular format converters included in the standard component collection of Pipeline  
223 Pilot. Afterwards, the initial receptor and ligand files can be parameterized depending on the charge  
224 model used, hydrogen model, etc. and additional components (pH, tautomers, etc.) can also be easily

225 included in the pipeline. Once the molecules are ready for the HTVS calculations, the docking parameters  
226 (degree of flexibility, simulation length, physical model, etc.) and parallel calculation parameters (batch  
227 size, number of processors to use, etc.) are also specified at the beginning of the third pipeline. In any  
228 case, the protocol also provides default parameters for all the components, so that the end user only needs  
229 to select ligand, receptor and binding site parameters to run FlexScreen calculations.



**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.

### 230 **Data Analysis from Virtual Screening Calculations**

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

### 237 **SOAP Implementation of FlexScreen**

238 The integration in Pipeline Pilot alone, or in other words, the use of Pipeline Pilot on just a desktop  
239 machine is, however, insufficient for really large in-silico screening campaigns. The improved accuracy  
240 of FlexScreen comes at the price of the computation cost of the underlying biophysical model. Therefore,  
241 we have implemented the FlexScreen Pipeline Pilot modules as a SOAP-based (Snell et al., 2002) client-  
242 service pair capable to operate on distributed architectures such as computing grids and clouds. We  
243 have developed a SOAP-based web service for the remote FlexScreen application using software such

---

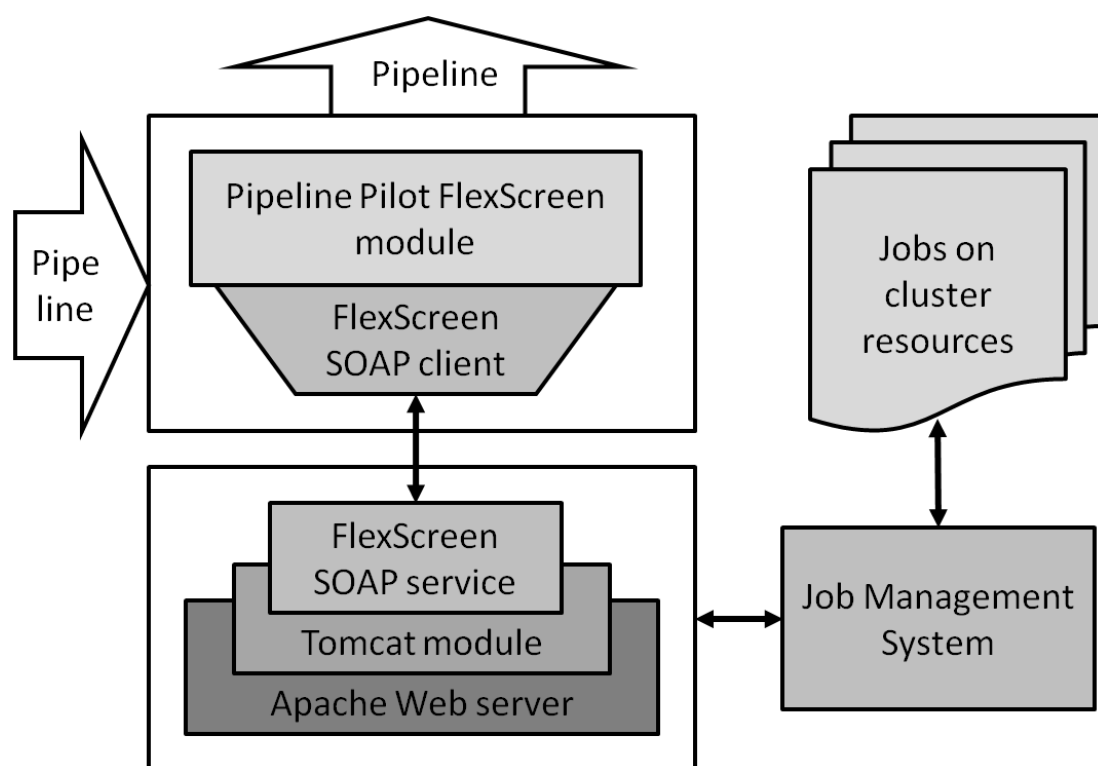
**Algorithm 1** Pseudocode of the FlexScreen workflow using SOAP services with Pipeline Pilot

---

```
1: for  $i = 1$  to numberofbatches do  
2:   SendLigandData  
3:   SendReceptorData  
4:   SendSimulationParameters  
5: end for  
6: for  $i = 1$  to numberofbatches do  
7:   ExecuteSimulationUnits  
8: end for  
9: ReceiveResultsData  
10: ReportResultsData
```

---

244 as Apache/Tomcat (<http://tomcat.apache.org>) or the Perl SOAP::Lite module (<http://soaplite.com>). The  
245 SOAP server contains sufficient processing functionality to perform the following tasks (cf. Fig.2):



**Figure 2. Architecture of the implemented FlexScreen module (cf. Fig.1).** This figure represents the case of use of distributed HPC resources via a SOAP client-server pair.

246 1. Receive a batch of ligands and receptor file as a SOAP message and save them to a file (steps 2  
247 and 3 of Algorithm 1). One of the advantages of using SOAP is that it allows a batch size to be specified,  
248 allowing the collation of a series of individual docking requests in a single request for efficiency.

249 2. Receive complementary information as SOAP messages (step 4 of Algorithm 1) and save it to files,  
250 e.g., protein active site, configuration files related to simulation parameters, etc.

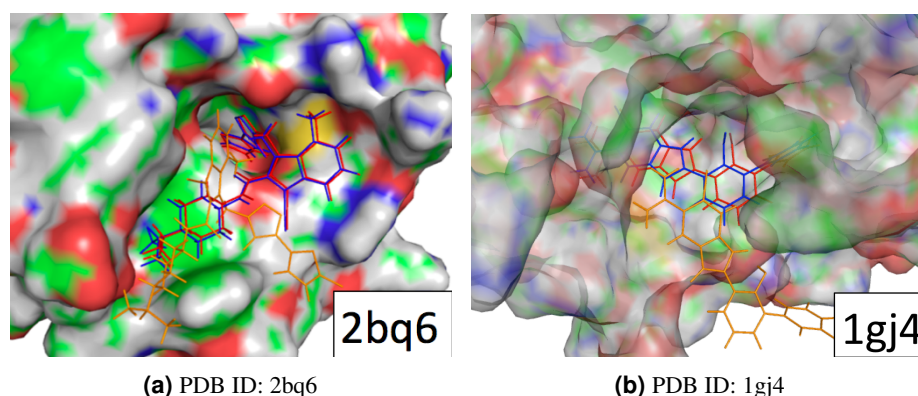
251 3. Generate and submit jobs to execute FlexScreen on HPC resources using the files previously created  
252 (step 7 of Algorithm 1).

253 4. Read the resulting files (step 9 of Algorithm 1) and pass them back as a SOAP message to the  
254 calling component. A report on the results will be automatically prepared (step 10 of Algorithm 1) as an  
255 interactive HTML report, a PDF document, or a spreadsheet.



### 256 Examples of Use

257 Results from a HTVS calculation performed by an end user are shown in Figs.3 and 4. As it can be seen  
 258 in Fig.4, the resulting data is clearly organized in tables which are directly opened in the web browser  
 259 after the screening calculations. The user can control the degree of detail in the final report interacting  
 260 with the “table parameters” component as well as reorganize easily and sort the final data with a few  
 261 mouse clicks in the web browser. There is also the possibility of exporting the results to other standard  
 262 formats, i.e., PDF, Word, Excel spreadsheets, CSV text files, etc. The end user can also obtain detailed  
 263 information about the 3D structure of the docked receptor–ligand conformations as can be seen in Fig.3,  
 264 very useful for compound optimization, posterior screenings, etc.



**Figure 3. 3D representation of the HTVS results obtained for two different receptor-ligand pairs.** 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

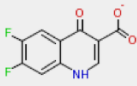
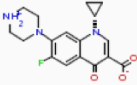
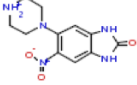
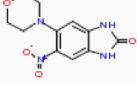
265 From the perspective of users’ experience, we found that the access to well-developed and validated  
 266 workflows using FlexScreen encourages the user to test and explore new ideas. Informal discussions with  
 267 users who have performed HTVS calculations with FlexScreen in this way confirms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

### 271 Integration of the FlexScreen Services in Further Science Gateways

272 In the last four years a few new science gateways have been developed or existing ones have been  
 273 extended to support the HTVS user community, e.g. MoSGrid (Krüger et al., 2014) developed on top of  
 274 WS-PGRADE or KNIME. To allow the reusability of services in the users’ preferred virtual environment,  
 275 we investigated the possibilities to integrate the FlexScreen services in the context of further science  
 276 gateways.

277 Since the FlexScreen services are SOAP based, a crucial prerequisite is the support of such services in  
 278 the science gateway. Furthermore, the science gateway needs to be workflow enabled for the different  
 279 tasks accomplished by each of the services to provide the whole pipeline of analysis steps. Since users  
 280 may have established preparation and post processing steps for the HTVS pipeline, another prerequisite  
 281 for considering a science gateway is the possibility to configure the execution environment of tasks in a  
 282 workflow or pipeline independent from each other. In our investigation we considered four workflow-  
 283 enabled science gateways widely used in the biomedical community.

284 WS-PGRADE (Kacsuk et al., 2012) is the flexible web user interface of the workflow system gUSE,  
 285 which supports the management of DAG-based workflows. The control structure is defined by data  
 286 dependencies and parameter sweep mechanisms allow for emulating loops over a defined range of  
 287 parameters and data. Each task in a workflow is represented by a job with input and output datasets and  
 288 each job can be configured for exploiting a resource independent of the configuration of dependent jobs.  
 289 Thus, a job can be configured as SOAP web service and connected with jobs defined for applying local,  
 290 cluster, grid and cloud resources or another SOAP web service. Thus, users can reuse the FlexScreen  
 291 services in an intuitive way.

DOCKING RESULTS OF streptavidin			
NAME	MOLECULE	AFFINITY	RMSD
jnk.vs.1-81_1		-102.113727	1.803142
jnk.vs.1-82_1		-84.001330	1.882979
jnk.vs.1-83_1		-34.849769	1.812568
jnk.vs.1-84_1		-65.361482	2.092396

**Figure 4.** 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 Fig.3.

292 The concept behind Galaxy (Goecks et al., 2010; Blankenberg et al., 2010; Giardine et al., 2005)  
 293 differs from WS-PGRADE but it also offers an intuitive web user interface with workflow management  
 294 capabilities for DAG-based workflows. It is designed as a tool box for intuitively creating and invoking  
 295 workflows with pre-configured tools in local, cluster and cloud environments. The administrator of a  
 296 Galaxy instance can configure SOAP web services, which are then available to the users (Wang et al.,  
 297 2009). Hence, users are able to integrate the FlexScreen services in their workflows.

298 Taverna (Wolstencroft et al., 2013) follows a different approach on the client side compared to WS-  
 299 PGRADE and Galaxy and the workbench needs to be installed by the users. Despite this drawback on the  
 300 users' side, it is widely adopted in the community. It supports besides DAG-based workflows also loops  
 301 as workflow constructs and is especially based on configuring each step in a workflow as SOAP-based  
 302 service. It is an ideal candidate for reusing the FlexScreen services.

303 While KNIME (Berthold et al., 2008) is also an easy-to-use workbench, which has to be installed  
 304 by the users, it supports command line tools and SOAP-based web services via its Generic Webservice  
 305 Client (<https://tech.knime.org/webservice-client>). KNIME is especially user-friendly, has rich workflow  
 306 management features and offers pre-configured packages. A user can easily integrate the FlexScreen  
 307 services into the workbench.

308 These four examples prove that the FlexScreen services are not only applicable in the native Pipeline  
 309 Pilot environment but also in other science gateways and, thus, reusable for a large user community  
 310 employing diverse science gateways for their research topics. The services can be connected with other  
 311 tools and services to improve the user experience on accomplishing their research in one user interface.

## 312 CONCLUSIONS AND FUTURE WORK

313 In this paper, we have considered the general problem of subject adaptation of existing IT technologies,  
 314 its customization for known domain of expertise of end users. We have described the implementation of a  
 315 HTVS methodology in a science gateway environment making use of the workflow environment provided

316 by Pipeline Pilot. The solution basing on SOAP and web services enables the exploitation of distributed  
317 HPC resources using a grid computing strategy.

318 From our point of view, the main drawback of Pipeline Pilot is that a yearly paid license is required.  
319 Therefore, not all research institutions would be able to cover these costs. It seems that open source  
320 alternatives to Pipeline Pilot exist, such as UNICORE, Kepler and Taverna, but we are not sure yet whether  
321 they offer the same or similar alternative. Thus, we will explore them in further studies.

322 Currently, we are also developing improved GPU-based versions of FlexScreen (Guerrero et al., 2011;  
323 Sánchez-Linares et al., 2011b,a) and planning its deployment on grid resources.

## 324 REFERENCES

- 325 Berthold, M., Cebon, N., Dill, F., Gabriel, T., Kötter, T., Meinel, T., Ohl, P., Sieb, C., Thiel, K., and  
326 Wiswedel, B. (2008). Knime: The konstanz information miner. In Preisach, C., Burkhardt, H.,  
327 Schmidt-Thieme, L., and Decker, R., editors, *Data Analysis, Machine Learning and Applications*,  
328 *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 319–326. Springer Berlin  
329 Heidelberg.
- 330 Blankenberg, D., Kuster, G. V., Coraor, N., Ananda, G., Lazarus, R., Mangan, M., Nekrutenko, A., and  
331 Taylor, J. (2010). *Galaxy: A Web-Based Genome Analysis Tool for Experimentalists*, chapter 19, pages  
332 Unit 19.10,11–12. John Wiley & Sons, Inc.
- 333 Catlett, C. (2002). The philosophy of TeraGrid: building an open, extensible, distributed TeraScale facility.  
334 In *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*  
335 *(CCGRID2002)*.
- 336 Catlett, C. (2005). TeraGrid: A Foundation for US Cyberinfrastructure. In Jin, H., Reed, D., and Jiang,  
337 W., editors, *Network and Parallel Computing*, volume 3779 of *Lecture Notes in Computer Science*,  
338 page 1. Springer.
- 339 Deelman, E., Gannon, D., Shields, M., and Taylor, I. (2009). Workflows and e-science: An overview of  
340 workflow system features and capabilities. *Future Generation Computer Systems*, 25(5):528 – 540.
- 341 Eisenberg, D. and McLachlan, A. D. (1986). Solvation energy in protein folding and binding. *Nature*,  
342 319(6050):199–203.
- 343 Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P.,  
344 Knoll, E. H., Shelley, M., Perry, J. K., Shaw, D. E., Francis, P., and Shenkin, P. S. (2004). Glide: A New  
345 Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy.  
346 *Journal of Medicinal Chemistry*, 47(7):1739–1749.
- 347 Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., Zhang, Y., Blankenberg,  
348 D., Albert, I., Taylor, J., Miller, W., Kent, W. J., and Nekrutenko, A. (2005). Galaxy: A platform for  
349 interactive large-scale genome analysis. *Genome Research*, 15(10):1451–1455.
- 350 Goecks, J., Nekrutenko, A., and Taylor, J. (2010). Galaxy: a comprehensive approach for supporting  
351 accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*,  
352 11.
- 353 Guerrero, G., Pérez-Sánchez, H., Wenzel, W., Cecilia, J. M., and García, J. M. (2011). Effective  
354 parallelization of non-bonded interactions kernel for virtual screening on gpus. In *5th International*  
355 *Conference on Practical Applications of Computational Biology; Bioinformatics (PACBB 2011)*,  
356 volume 93, pages 63–69. Springer.
- 357 Halgren, T. A., Murphy, R. B., Friesner, R. A., Beard, H. S., Frye, L. L., Pollard, W. T., and Banks, J. L.  
358 (2004). Glide: A New Approach for Rapid, Accurate Docking and Scoring. 2. Enrichment Factors in  
359 Database Screening. *Journal of Medicinal Chemistry*, 47(7):1750–1759.
- 360 Irwin, J. J. and Shoichet, B. K. (2005). ZINC—a free database of commercially available compounds for  
361 virtual screening. *Journal of Chemical Information and Modeling*, 45(1):177–182.
- 362 Jacob, R. B., Andersen, T., and McDougal, O. M. (2012). Accessible high-throughput virtual screening  
363 molecular docking software for students and educators. *PLoS computational biology*, 8(5):e1002499.
- 364 Kacsuk, P., Farkas, Z., Kozlovsky, M., Hermann, G., Balasko, A., Karoczkai, K., and Marton, I. (2012).  
365 Ws-pgrade/guse generic dcii gateway framework for a large variety of user communities. *Journal of*  
366 *Grid Computing*, 10(4):601–630.
- 367 Kokh, D. B. and Wenzel, W. (2008). Flexible side chain models improve enrichment rates in in silico  
368 screening. *Journal of Medicinal Chemistry*, 51(19):5919–5931. PMID: 18771256.
- 369 Krüger, J., Grunzke, R., Gesing, S., Breuers, S., Brinkmann, A., de la Garza, L., Kohlbacher, O., Kruse, M.,

- 370 Nagel, W. E., Packschies, L., Müller-Pfefferkorn, R., Schäfer, P., Schärfe, C., Steinke, T., Schlemmer,  
371 T., Warzecha, K. D., Zink, A., and Herres-Pawlis, S. (2014). The mosgrid science gateway – a complete  
372 solution for molecular simulations. *Journal of Chemical Theory and Computation*, 10(6):2232–2245.
- 373 Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J., and Zhao,  
374 Y. (2006). Scientific workflow management and the kepler system. *Concurrency and Computation:  
375 Practice and Experience*, 18(10):1039–1065.
- 376 Meng, E., Shoichet, B., and Kuntz, I. (1992). Automated Docking with Grid-Based Energy Evaluation.  
377 *Journal of Computational Chemistry*, 13(4):505–524.
- 378 Merlitz, H. and Wenzel, W. (2002). Comparison of stochastic optimization methods for receptor–ligand  
379 docking. *Chemical Physics Letters*, 362(3-4):271–277.
- 380 Merlitz, H. and Wenzel, W. (2004). High throughput in-silico screening against flexible protein receptors.  
381 In Lagana, A., editor, *Proceedings of the International Conference on Computational Science and its  
382 Applications, Assisi, Italy (ICCSA 2004)*, volume 3045 of *Lecture Notes in Computer Science*, page  
383 465. Springer.
- 384 Morris, G. M., Goodsell, D. S., Huey, R., and Olson, A. J. (1996). Distributed automated docking  
385 of flexible ligands to proteins: Parallel applications of autodock 2.4. *Journal of Computer-Aided  
386 Molecular Design*, 10(4):293–304.
- 387 Pandey, S., Voorsluys, W., Rahman, M., Buyya, R., Dobson, J., and Chiu, K. (2009). A grid workflow  
388 environment for brain imaging analysis on distributed systems. *Concurrency and Computation: Practice  
389 and Experience*, 21(16):2118–2139.
- 390 Pérez-Sánchez, H., Kondov, I., Garcia, J., Klenin, K., and Wenzel, W. (2011). A pipeline pilot based soap  
391 implementation of flexscreen for high-throughput virtual screening. In Terstyánszky, G. and Kiss, T.,  
392 editors, *IWSG-Life*, volume 819 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- 393 Pérez-Sánchez, H. and Wenzel, W. (2011). Optimization methods for virtual screening on novel computa-  
394 tional architectures. *Current computer-aided drug design*, 7(1):44–52.
- 395 Sánchez-Linares, I., Pérez-Sánchez, H., and García, J. M. (2011a). Accelerating grid kernels for virtual  
396 screening on graphics processing units. In *Proceedings of the 14th International Conference on Parallel  
397 Computing - ParCo 2011*, ParCo 2011.
- 398 Sánchez-Linares, I., Pérez-Sánchez, H., Guerrero, G. D., Cecilia, J. M., and García, J. M. (2011b).  
399 Accelerating multiple target drug screening on gpus. In *Proceedings of the 9th International Conference  
400 on Computational Methods in Systems Biology*, CMSB '11, pages 95–102, New York, NY, USA. ACM.
- 401 Sild, S., Maran, U., Romberg, M., Schuller, B., and Benfenati, E. (2005). Openmolgrid: Using automated  
402 workflows in grid computing environment. In Sloot, P. M. A., Hoekstra, A. G., Priol, T., Reinefeld, A.,  
403 and Bubak, M., editors, *EGC*, volume 3470, pages 464–473. Springer.
- 404 Snell, J., Tidwell, D., and Kulchenko, P. (2002). *Programming Web Services with SOAP*. O'Reilly &  
405 Associates, Inc., Sebastopol, CA, USA.
- 406 Wang, R., Brewer, D., Shastri, S., Swayampakula, S., Miller, J., Kraemer, E., and Kissinger, J. (2009).  
407 Adapting the galaxy bioinformatics tool to support semantic web service composition. In *Services - I,  
408 2009 World Conference on*, pages 283–290.
- 409 Wenzel, W. and Hamacher, K. (1999). Stochastic tunneling approach for global minimization of complex  
410 potential energy landscapes. *Phys. Rev. Lett.*, 82:3003–3007.
- 411 Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop,  
412 I., Nenadic, A., Fisher, P., Bhagat, J., Belhajjame, K., Bacall, F., Hardisty, A., Nieva de la Hidalgo, A.,  
413 Balcazar Vargas, M. P., Sufi, S., and Goble, C. (2013). The taverna workflow suite: designing and  
414 executing workflows of web services on the desktop, web or in the cloud. *Nucleic Acids Research*,  
415 41(W1):W557–W561.
- 416 Yang, X., Bruin, R., and Dove, M. (2010). Developing an end-to-end scientific workflow: A case  
417 study using a comprehensive workflow platform in e-science. *Computing in Science Engineering*,  
418 12(3):52–61.