e-Science Central: Cloud-based e-Science and its application to chemical property modelling

Hugo Hiden, Paul Watson, Simon Woodman and David Leahy
e-Science Central: Cloud-based e-Science and its application
to chemical property modeling

H. Hiden, P. Watson, S. Woodman and D. Leahy

Abstract

This paper describes the design of a cloud computing platform - e-Science Central (e-SC) - which provides both Software and Platform as a Service for scientific data management, analysis and collaboration. e-SC can be deployed on both private (e.g. Eucalyptus) and public Clouds (Amazon AWS and Microsoft Windows Azure). The SaaS application allows scientists to upload data, edit and run workflows, and share results in the cloud. It is underpinned by a scalable cloud platform consisting of a set of services designed to support the needs of scientists. The platform is exposed to developers so that they can easily upload their own analysis services into the system and make these available to other users. A REST-based API is also provided so that external applications can leverage the platform’s functionality. This paper describes the design of e-SC, focusing on its exploitation of clouds. Its usage is demonstrated by a description of a recent project in which it was used for chemical modeling by cancer researchers.
Bibliographical details

HIDEN, H., WATSON, P., WOODMAN, S., LEAHY, D.

e-Science Central: Cloud-based e-Science and its application to chemical property modelling
[By] H. Hiden, P. Watson, S. Woodman and D. Leahy

(University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-1227)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE

Abstract

This paper describes the design of a cloud computing platform - e-Science Central (e-SC) - which provides both Software and Platform as a Service for scientific data management, analysis and collaboration. e-SC can be deployed on both private (e.g. Eucalyptus) and public Clouds (Amazon AWS and Microsoft Windows Azure). The SaaS application allows scientists to upload data, edit and run workflows, and share results in the cloud. It is underpinned by a scalable cloud platform consisting of a set of services designed to support the needs of scientists. The platform is exposed to developers so that they can easily upload their own analysis services into the system and make these available to other users. A REST-based API is also provided so that external applications can leverage the platform’s functionality. This paper describes the design of e-SC, focusing on its exploitation of clouds. Its usage is demonstrated by a description of a recent project in which it was used for chemical modeling by cancer researchers.

About the authors

Hugo Hiden is the software development lead on the SIDE project. His main interests are the development of the e-Science Central system, cloud based data analysis systems and workflows. Before this, he was the Technical Director of the UK North Eastern Regional e-Science Centre (NEReSC) at Newcastle University where his primary responsibility was the technical co-ordination of the various research projects within the centre. Prior to this, he spent 6 years in industry developing advanced data integration, analysis and modeling tools at Avantium Technologies and GSE Systems Inc. He holds a PhD in the application of Genetic Programming to chemical process data analysis and has published numerous papers on the subjects of process monitoring, computer security and collaborative R&D.

Paul Watson is Professor of Computer Science, Director of the Informatics Research Institute, and Director of the North East Regional e-Science Centre. He also directs the UKRC Digital Economy Hub on “Inclusion through the Digital Economy”. He graduated in 1983 with a BSc (I) in Computer Engineering from Manchester University, followed by a PhD in 1986. In the 80s, as a Lecturer at Manchester University, he was a designer of the Alvey Flagship and Esprit EDS systems. From 1990-5 he worked for ICL as a system designer of the Goldrush MegaServer parallel database server, which was released as a product in 1994. In August 1995 he moved to Newcastle University, where he has been an investigator on research projects worth over 20M. His research interests are in scalable information management. Paul Watson teaches information management on the System Design for Internet Applications Msc and the e-business MSc. In total, Paul Watson has over forty refereed publications, and three patents. Professor Watson is a Chartered Engineer, a Fellow of the British Computer Society, and a member of the UK Computing Research Committee.

Simon Woodman is a Research Associate in the Scalable Information Management group. His interests include workflow enactment, especially those systems that allow distributed coordination; service description languages, data provenance, social networking and collective intelligence. He is currently developing an e-Science system to allow secure data sharing and workflow enactment for the neuroscience domain. Simon was awarded his PhD in 2008 for his thesis on “A Programming System for Process Coordination in Virtual Organisations”, supervised by Prof. Santosh Shrivastava. During his undergraduate studies, Simon worked at IBM Hursley for a year as part of a team developing the CICS Information Centre. Simon received his BSc (hons) from Newcastle in 2002.

Prof. David Leahy was a Senior Executive with what is now AstraZeneca Pharmaceuticals before leaving to found Cyprotex, an information company supporting research by 150 Pharmaceutical companies world-wide. Having led Cyprotex through start-up, MBO and IPO on the London Stock Exchange he left the company to work on new business ventures and academic research at Newcastle University, UK as part of the Newcastle Science City initiative. He held a "Professor of Practice" position in the Business School and had a research group in Computer-Aided Molecular Design funded by Cancer Research UK. He has been active in scientific software development applied to drug discovery as a developer, manager and entrepreneur for over 25 years.
Suggested keywords

E-SCIENCE
IAAS
PAAS
CLOUD COMPUTING
e-Science Central

Cloud-based e-Science and its application to chemical property modelling

Hugo Hiden, Paul Watson, Simon Woodman, David Leahy
School of Computing Science
Newcastle University
Newcastle-upon-Tyne, UK

Abstract- This paper describes the design of a cloud computing platform - e-Science Central (e-SC) - which provides both Software and Platform as a Service for scientific data management, analysis and collaboration. e-SC can be deployed on both private (e.g. Eucalyptus [1]) and public Clouds (Amazon AWS [2] and Microsoft Windows Azure [3]). The SaaS application allows scientists to upload data, edit and run workflows, and share results in the cloud. It is underpinned by a scalable cloud platform consisting of a set of services designed to support the needs of scientists. The platform is exposed to developers so that they can easily upload their own analysis services into the system and make these available to other users. A REST-based API is also provided so that external applications can leverage the platform’s functionality. This paper describes the design of e-SC, focusing on its exploitation of clouds. Its usage is demonstrated by a description of a recent project in which it was used for chemical modeling by cancer researchers.

I. INTRODUCTION

Cloud computing has the potential to revolutionize e-science by giving scientists the computational resources they need, when they need them. On its own however, clouds do not make it easier to design, implement and maintain the often complex, scalable, dependable applications needed to support science. The problems can be seen when the various types of Cloud computing offerings currently available are considered [4]:

Infrastructure as a Service (IaaS): This is typical of many Cloud offerings, for example Amazon EC2 [2]. Using IaaS, developers can dynamically provision compute and storage resources, and they typically have control over the whole software stack including the operating system. The drawback is that, for the majority of potential scientific users, access to raw hardware is of little use as they lack the skills and resources needed to design, develop and maintain the robust, scalable application they require.

Platform as a Service (PaaS): Provides a higher-level of abstraction than IaaS as developers are provided with a platform containing services that can be used to build applications. For example, force.com [5] provides a variety of hosted services which can be used to develop CRM-related applications in the cloud. The drawback is that, for obvious reasons, current platforms focus on services required for business applications, rather than those needed for scientific data storage and analysis (in Section II we describe our view of the platform services that are needed, based on our experiences in working with a wide range of scientists over the past 10 years).

Software as a Service (SaaS): Makes packaged applications available to users through the Web. Examples include Google Docs and salesforce.com [6]. Again, the problem is that the applications provided to date have focused on the large, commercial markets such as e-mail and document management. Some of these functions may be useful to scientists (for instance Google Charts), but they do not meet the majority of the needs of scientists.

As a result of these limitations, and our experience of the first decade of e-science, we have concluded that there will be relatively few science research groups with the skills and available effort required to build scalable, dependable science applications on the existing Cloud offerings. The danger is therefore that the potential of the cloud to revolutionize e-science will not be fully realized.

To address this, we have designed e-Science Central (e-SC), a cloud-based Science Platform as a Service that allows scientists to store, analyze and share data in the cloud. The e-SC design was influenced by our work over the past decade on 25 e-science projects, collaborating with researchers from ten different disciplines, at the North East Regional e-Science Centre. e-SC has now been in constant use for over 18 months, with over 100 users. Over 200,000 workflow executions have taken place.

The paper’s main contributions are to describe:

1. The platform’s main cloud services, which are: data storage, service execution, security, workflow enactment and provenance. These have been designed to be as independent as possible of any specific cloud infrastructure, and e-SC can run on both private clouds (e.g. Eucalyptus [1]) and public clouds (e.g. Amazon EC2 [2], and Microsoft Windows Azure [3]).
2. The environment and tooling provided to allow users to develop and upload new services to run on the cloud platform.
3. The Software as a Service (SaaS) interface that allows users to access the cloud platform’s services entirely through a web browser.
4. An case study in which e-SC was used by cancer researchers to reduce the estimated time for building predictive models of chemical activity from 5 years to 3 weeks.

This paper is structured as follows: Section II describes the overall functionality of e-SC through its Science Software as Service Web 2.0 interface; Section 0 then describes the key
platform services; finally, Section IV demonstrates the use of the platform for a chemical modeling application.

II. E-SCIENCE CENTRAL OVERVIEW

This section describes the overall functionality of e-SC as exposed through its Science Software as Service Web 2.0 interface (www.esciencecentral.co.uk). Our work with a wide range of scientists over the past ten years has identified four key requirements that users require from an e-science platform: they need to store data, analyze it, automate the analysis, and share data in a controlled way. The e-SC “Software as a Service” application allows users to do this entirely through a web browser. We chose to offer this “Web only” approach after investigating reasons for the lack of uptake of some existing e-science tools for university research, and industry R&D – it became clear that many groups do not have the skills or resources needed to deploy and maintain applications, while many organizations actually prevent users from deploying their own software. Further, the browser-only approach suits the new way of working for many scientists, who wish to do their work wherever they are, on a mixture of systems ranging from mobile devices, laptops and desktop PCs, both at home and at work.

After having logged in, users are presented with the screen such shown in Figure 1.

![e-Science Central Front Page](image)

**Figure 1.** e-Science Central Front Page

This front page displays updates and new content, as well as new social network connections. Users can search for users and connect to them; they can also join existing groups and create new ones. As will be seen, this forms the basis of security.

Users can upload experimental data into the system. For IP protection, all data is versioned, linked to the digital signature of the user, and time-stamped. (Figure 2 shows the page describing one such data item).

The default security for data is that it is private to the owner (e.g. the user who uploaded it), but at any time the owner can choose to share the data with others to whom they are connected through the social network, or to everyone in groups that they are members of, or to make it public. Public data is reachable through web search engines, so satisfying a common requirement from institutions and funding bodies that selected data should be openly available to encourage dissemination and support “open innovation” [7].

![Uploaded data](image)

**Figure 2.** Uploaded data

All data can have associated Metadata which allows it to be described and discovered by the e-Science Central search engine. Two options are supported:

- **Tagging:** Users can add arbitrary tags to describe their data.
- **Metadata documents:** Users can upload metadata documents and associate these with data. This allows, for example, structured XML files to be used to describe data in accordance with a pre-defined schema. Users can then use XPath [8] to issue structured searches for data. To remove the need for users to construct XML documents and queries manually, custom form-based interfaces can be added as applications that use the e-Science Central API (Section 0).

Whilst browsing or searching metadata provides one way for users to find data that is of interest to them, e-Science Central offers “Blogs” in which scientists can record and publish their work. Blog entries are under the same security regime as all data in the system, and so users can choose who to share their contents with. A unique feature is that blogs describing the results of data analysis can link to the data and the workflow, so allowing others to check the process by which results were created, to analyze the data using other methods, and to re-use the same workflows on other data.

Figure 2 also shows that e-SC displays the provenance of each piece of data. This show how it was created (as the result of an analysis or user upload) and its subsequent history.

A key feature of e-Science Central is that it allows users to analyze data, rather than just to share it as is the case with many collaboration systems. Its in-browser, workflow editor (Figure 3), written using html5, allows users to build a workflow by dragging services from the structured list on the left of the screen, and then connecting them together. The user...
can then click to run the workflow, and the results are then displayed at the bottom of the screen.

![Workflow Editor](image1.png)

**Figure 3.** The html5 workflow editor

All the functionality of the SaaS application described in this section is provided by the underlying, scalable science cloud platform which is now described.

III. **THE E-SCIENCE CENTRAL SCIENCE CLOUD PLATFORM**

The scientific platform offered by e-SC gives science application developers a scalable, high-level, cloud-based platform on which to build their applications. The platform architecture, including the main services within it, are shown in Figure 4. These build on processing and storage provided by an underlying Infrastructure as a Service cloud. As will be described, each service has a virtualized interface to the infrastructure, so that the platform is not tied to any one cloud provider: the platform has been run on a private Eucalyptus cloud [1] and on Amazon AWS [2]. In Section 0 we describe a case study in which the execution of services within a workflow runs on Microsoft Windows Azure [3].

While the platform services are made available to users through the SaaS application described in Section II, as shown in Figure 4, programmatic access is also provided through a REST API (with Java and .Net clients), so allowing applications to build directly on the cloud platform. This has been used to build a variety of desktop and mobile applications that rely on e-SC to store and process data in the cloud. Each of the platform services is now discussed.

A. **Data Storage**

e-SC stores user data, services and workflows in the cloud. All data is accessed through a virtualized storage system that provides a level of independence from the underlying cloud infrastructure - drivers are currently available for local filesystems and the Amazon S3 service.

All data is versioned: when a file is stored, if a previous version exists, then a new version is automatically created. This is important for allowing users to work with old versions of data, services and workflows, so supporting the reproduction of experiments, and investigations into the effects of changes to data and the analysis process over time.

![Science Cloud Platform](image2.png)

**Figure 4.** Science Cloud Platform

All versions are signed, so that it is possible to reliably validate data and detect any modifications. Public-Private key cryptography and x509 certificates [9] are used for this; a set of keys and certificates is created for each user during the registration process and these are subsequently used to sign all data stored in the system. They allow the authenticity of any data stored in e-Science Central to be verified by users and applications.

B. **Security**

Scientists in both industry and academia require the ability to protect their data and analysis processes until they chose to publish. e-SC therefore gives users fine-grained security control. Experiment [10] has demonstrated the potential of social networking to support collaboration in scientific research and so e-SC also provides this, allowing users to connect to each other, and form groups. These connections are then used as the basis for users to control the security of all information, including data, services, blogs and workflows.

Data security has been implemented in the e-SC platform, rather than depending on the capabilities of the underlying storage system. This allows a range of underlying cloud storage services to be used, with no dependencies on the security mechanisms they natively offer.

All entities in the system (users, groups, data, workflows and services) are represented internally as subclasses of a single object. This allows them all to be guarded by a single security mechanism which checks every access. Access Control Lists (ACLs), associated with each object, are respected by this core object access code. These express security policies in terms of the actions that specific users and groups are allowed to perform on that resource. The actions can be: Read, Write, Delete and Add (allowing the addition of data to a container resource e.g. a Folder). The ACLs are stored in a
database as triples comprising: the identity of the resource, the identity of the user or group and the action permitted. The architecture for this feature has been adopted from our earlier GOLD project [11].

C. Services

e-SC allows users to upload services (e.g. for data analysis) into the platform. They are stored in the e-SC data store and dynamically deployed in the cloud when called within a workflow enactment. The security infrastructure allows users to share these services with other individuals or groups, or alternatively they can be kept private.

Services can be developed using a number of languages including Java, R [12] and Octave [13], and work is under way to add support for Python, Perl and .NET. Services are packaged and uploaded using a desktop service development tool that is made available to developers (Figure 5).

![Figure 5. The Service Development Tool](image)

Services access the platform services through the same API that is made available to external applications (as described above). There is therefore a common way of interacting with the system regardless of whether code is executing as a service in a workflow, or in an external application.

D. Workflow Enactment

The Workflow Engine provides the means to scalably execute code in e-SC. It differs from engines such as Taverna [14] and Kepler [15] in that it automatically deploys services on multiple nodes in the cloud as required to provide scalability in workflow execution throughout.

Workflows will have been defined graphically using the SaaS workflow editor (Figure 3) or within the service development environment (Figure 5). To build a workflow, services are dragged from a palette (which contains all of the available services that the given user has access to) onto a workflow canvas. These services contain input and output ports, which can then be linked together. The input and output ports are able to restrict the types of data that can be sent to them, meaning that only compatible ports can be connected. Currently, the e-SC workflow engine supports the data types for ports shown in Table 1.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data-wraper</td>
<td>A rectangular grid of data. Data is typically arranged with each column representing a single attribute, with instances of the attributes arranged in rows. The data-wraper type offers the option to stream large data sets through services in chunks of rows. CSV files are a typical example of this type of data.</td>
</tr>
<tr>
<td>file-wraper</td>
<td>A reference to a file, or list of files. The internal structure of the data is opaque to the workflow system, so no streaming is possible. Interpretation of the data is left entirely to the service code. An example of this type of data would be an image or text file.</td>
</tr>
<tr>
<td>object-wraper</td>
<td>An arbitrary, serialized Java object that can contain parameters or sets of parameters. These are usually used to pass complex objects (for example entire models, or sets of scaling parameters) between services.</td>
</tr>
</tbody>
</table>

Table 1. Workflow data types

In the current implementation, all the workflow services within a single invocation of a workflow execute on the same cloud node, and so the intermediate data is passed between them via a temporary workflow invocation folder held on the filesystem of that node. To provide scalable performance, instances of the workflow engine are deployed on each of a set of cloud nodes, with each enacting different workflows.

The workflow enactment process is shown in Figure 6, and is now described.

![Figure 6. Workflow Execution](image)

The first step of the process (1) creates a simple workflow “Start” message which contains the ID of the workflow and a security token defining the user. This message is then placed onto a standard JMS (Java Message Service) queue where it
waits for the next available workflow enactment node to retrieve it. When the message is retrieved by an enactment node (2), the node first uses the workflow ID to download the full workflow definition from e-SC storage via the API (3). This workflow definition is then passed to the Workflow Execution Engine (4) where the execution path through the workflow is analysed to produce a Message Plan which describes the order in which each service within the workflow must be executed. The messages in this Message Plan are then sent sequentially to the Message Execution Engine (5) which is responsible for actually executing the service.

Once a message reaches the Message Execution Engine, one of two sequences of events occur. If the software required to provide the service is already installed in the local Service Library, the service is executed to process the message. Alternatively, if this software has not already been downloaded to the enactment node, the standard service package (as previously created and uploaded by the service development tool – Figure 5) is downloaded from e-SC storage and installed in the local Service Library. The service is then executed to process the message. This is repeated for each step in the workflow until all of the messages have been executed (6). At this time, a report describing the outcome of the workflow is sent back to e-SC via the API. This report contains details of each message execution, along with any data emitted on Standard Output and Standard Error streams of the processes that processed the individual workflow service messages.

E. Provenance

As e-SC supports the storage, analysis and sharing of data, there is the opportunity to collect very rich provenance information. The e-SC provenance service therefore stores a database of all system events such as data access, workflow execution and interactions through the social networking service. A query interface can be used to extract information that is useful in different contexts. This is widely exploited in e-SC; for example the SaaS application described in Section II allows the user to view the history and life cycle of every piece of data within the system, including who created it, who downloaded it, what version of which services (in what workflow) accessed the data, and who the data has been shared with. This allows for example scientists to reproduce experiments, ascertain which files have been derived from a particular version of a service (perhaps now known to have a bug) and see who else has accessed files they have made available.

This provenance data model is based on the Open Provenance Model (OPM) Version 1.1 [16]. Data can be exported in the standard OPM format for analysis using tools such as the OPM toolbox, for example, to produce a directed acyclic graph of the history of an object. Objects in OPM are categorized as artifacts, processes or actors, which correspond to nodes within the graph. Vertices in the graph represent a causal dependency between two objects and are annotated with types such as wasGeneratedBy, used, wasControlledBy and wasDerivedFrom. A simple example, taken from the case study described in Section IV is shown in Figure 7. It describes how two models (PLS and Neural Network) were created. These models are generated by a model-building workflow which took as input a file containing chemical descriptor values, and was controlled (executed) by the user Hugo Hiden. A feature of OPM is that it allows for alternate views of how an artifact was generated. The white and gray sections of Figure 7 show, in differing levels of detail, how the two models were constructed. Whilst the white version treats the workflow as a single ‘black box’, the gray version of events breaks it into its constituent parts (simplified for this example).

Integration with the security service ensures that the level of detail that provided in the result of a provenance service query is dependent on the access privileges the user has for the constituent objects. For instance, if they have access to the workflow they can see the gray section of Figure 7, whereas if they do not, they will only see the white section. The effect of this is that if a user makes a data public, she will be able to see if workflows have used it as input, but will only be able to see what those workflows do if their owner has made them public too.

In this section we have described the key services of the e-SC cloud platform for science. In the next we show their use through a case study.

IV. CASE STUDY: CHEMICAL MODELING

This section of the paper describes a case study whereby e-SC was integrated into an existing scientific application that builds models of chemical properties. The aim was to speed-up the application so that it could process the vast amounts of new chemical activity data that had recently been published, in a reasonable time. The trigger for the work was an estimate that
on its existing single-server architecture, the application would take five years to process the new data.

The application was using Quantitative Structure-Activity Relationship (QSAR) [17] in order to identify new drugs that may block certain classes of cancer-causing kinase enzymes. QSAR quantitatively correlates the chemical structure of compounds with properties such as reactivity or biological activity. For example, as the number of carbon atoms in a hydrocarbon increases, so does its boiling point.

There are over 3000 descriptors that can be used to relate the quantifiable structural properties of a compound to its more complex properties such as biological activity. This, coupled with the wide range of different modeling algorithms, makes it computationally expensive to generate high-quality predictive models.

The chemists collaborating in this case study had developed a system to automate the process of building QSAR models which is referred to as the Discovery Bus [17]. It automatically generates new QSAR models as new data or modeling techniques are made available. This creates a continuously expanding library of predictive models that can be used to design new drugs. At the highest level, the operation of the Discovery Bus can be represented using the flow chart shown in Figure 8.

For these experiments, e-Science Central was configured to run the workflows concurrently on up to 100 Windows Azure nodes. Models were developed for the entire ChemBL database [18] with the following results (Table 2):

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total computation time</td>
<td>22 days</td>
</tr>
<tr>
<td>Total number of model generated</td>
<td>757,563</td>
</tr>
<tr>
<td>Number of stable and valid models</td>
<td>23,055</td>
</tr>
<tr>
<td>Number of biological targets with stable and valid models</td>
<td>3,011</td>
</tr>
<tr>
<td>Total data transferred (in/out)</td>
<td>16.74 B</td>
</tr>
</tbody>
</table>

Table 2. QSAR modelling results

Therefore, by using the cloud, an estimated 5 years of model-building time on the existing single Discovery Bus server was completed in a 3 week period. This is a greater than 100 times speed-up as each Azure node is more powerful than the existing Discovery Bus server on which the model-building had been run.

The competitive workflow pattern used in the QSAR model-building is extremely well matched to cloud computing. When new input data is available this can be processed in parallel on a set of nodes, each running a different model-building algorithm. Similarly, when a new model-building algorithm is designed, this can be run against the existing data, and its outputs compared with existing models to see if it can outperform the existing algorithms. Importantly, when there are no new data or algorithms, no cloud resources are needed, and so no costs are incurred.

V. CONCLUSIONS

This paper has described e-Science Central both in terms of its Science Platform as a Service, and its Software as a Service application. The aim has been to make it easier and for scientists to store, share and analyze their data, and for developers to create new scientific services and applications. We firmly believe that the design and deployment of a science cloud platform above the cloud infrastructure platforms that are now commercially available has been important for achieving these goals.

The Discovery Bus case-study has shown that the system can scale to a large number of nodes in order to process large amounts of computation that would otherwise not have been possible for our chemist collaborators. Reducing their Discovery Bus application run-time from five years to three weeks allowed them to generate many more, and better, models than would otherwise have been possible. It also provided a good driver to stress and guide the optimization of e-Science Central.

e-Science Central currently has over 100 users working on a range of sciences, and so the number of available workflows, services and datasets is constantly increasing. The platform itself is now stable, and so we are currently developing new tools that exploit its services. These include the mining of provenance data to provide advice to users, and tools to allow other applications to benefit from the competitive workflow pattern represented by the Discovery Bus application.

ACKNOWLEDGMENT

We would like to thank Microsoft for funding “Project Junior”, which focused on the Discovery Bus case study. The
One North East regional development agency and EPSRC (through the North East Regional e-Science Centre and the CARMEN project) funded the development of parts of e-Science Central. The authors would also like to thank Paul Appleby, Christophe Poulain and Savas Parastatidis of Microsoft for their support, and also Dominic Searson, Vladimir Sykora and Jacek Cala for their work in Junior.

REFERENCES


[18] EBI, "ChEMBL, www.ebi.ac.uk/chembl.db."