Deliverable D2.4

Final framework

Grant Agreement	Health-F5-2008-200787
Acronym	OpenTox
Name	An Open Source Predictive Toxicology Framework
Coordinator	Douglas Connect

SEVENTH FRAMEWORK PROGRAMME

Contract No.	Health-F5-2008-200787	
Document Type:	Deliverable Report	
WP/Task:	WP2	
Name	Final framework	
Document ID:	OpenTox Deliverable Report WP2-D2.4	
Date:	October 26, 2011	
Status:	Revised draft	
Organisation:	Ideaconsult Ltd, Sofia, Bulgaria	
Contributors:	Nina Jeliazkova	IDEA
	Vedrin Jeliazkov	IDEA
	Andreas Karwath	ALU-FR
	Martin Gütlein	ALU-FR
	Dmitry S. Druzhilovsky	IBMC
	Andreas Maunz	IST
	David Vorgrimmler	IST
	Chrisoph Helma	IST
	Rama Kaalia	JNU
	Surajit Ray	JNU
	Indira Ghosh	JNU
	Pantelis Sopasakis	NTUA
	Sunil Chawla	SL
	Fabian Buchwald	TUM
	Jörg Wicker	TUM
	Roman Affentranger	DC
	Nicki Douglas	DC
	Barry Hardy	DC

Distribution:	Public

Purpose of Document:	To document results for this deliverable

Document History:	1 – Initial draft distributed on Apr 30, 2011
	2 – Revised draft distributed on May 30, 2011
	3 – Revised draft distributed on May 31, 2011
	4 – Revised draft distributed on June 1, 2011
	5 – Revised draft distributed on June 7, 2011
	6 – Revised draft distributed on June 9, 2011
	7 – Revised draft distributed on June 15, 2011
	8 – Revised draft distributed on June 20, 2011
	9 – Revised draft distributed on August 18, 2011
	10 – Revised draft distributed on August 21, 2011
	11– Revised draft distributed on September 28, 2011
	12 – Revised draft distributed on October 7 2011
	13 – Revised draft October 30, 2011

Table of Contents

Table of Figures

Summary

The OpenTox predictive toxicology framework aims at providing a unified access to toxicity data and predictive models, as well as validation procedures. This is achieved by i) an information model based on a common ontology; ii) links to related ontologies; iii) data and algorithms, available through a standardized web services interface, where every compound, dataset or predictive method has a unique web address that can be used to retrieve its Resource Description Framework (RDF) representation or to initiate the associated calculations. The OECD principles of (Q)SAR validation, as well as the additional goals of software interoperability, flexibility, transparency, and extensibility, have been considered as major requirements in the design and implementation of the OpenTox framework, in full compliance with the REACH legislation and the corresponding user demand. This report details how these requirements were met in the final version of the OpenTox framework released during the FP7 project.

A key feature of the OpenTox framework is that it has been designed in a multi-domain friendly way, which is essential for data and model sharing, reproducibility and validation of prediction results. The framework includes web services for compounds, datasets, features, algorithms, models, ontology, validation, reporting, tasks, as well as authentication, authorisation and policy management. These services could be used as a distributed framework for processing of resource-intensive tasks and data sharing or in a fully independent way, according to the specific needs. The advantage of exposing the functionality via the OpenTox Application Programming Interfaces (APIs) is seamless interoperability, not only within a single web application, but also in a network of distributed services. The services provide a basis for building web mashups, end-user applications with friendly GUIs, as well as for embedding the functionalities in existing workflow systems.

Based on these web services 11 end-user oriented applications were created: a) ToxPredict, which predicts and reports on toxicities for endpoints for a user-provided input chemical structure, b) ToxCreate, which builds and validates a predictive toxicity model based on a user-provided input toxicology dataset, c) lazar, which is a stripped-down version of ToxCreate, d) Q-Edit, which facilitates the creation of QPRF reports, e) QMRF editor, which helps preparing models for submission in the QMRF inventory, f) CheS-Mapper, which makes it easy to look at complex datasets and intuitively provides important information to the user, g) IBMCDesc, which calculates multilevel and quantitative neighbourhoods of atoms descriptors, h) MaxToxMCSS, which is a suite of tools for model training and usage based on a maximum common substructure algorithm, i) ToxDesc, which calculates a range of different descriptors , j) Bioclipse-OpenTox set of plugins and workflows, which enables access to the OpenTox framework through the popular Bioclipse workbench for the life sciences, and k) Taverna workflows, which illustrate how the OpenTox framework can be accessed from the Taverna workflow management system.

This report includes short descriptions of the OpenTox web services, applications and workflows, as well as of their intended use. In addition, we briefly present the ToxOtis, OT-client, OT-sso, and opentox-ruby client software libraries, which ease the integration of 3rd party tools with the OpenTox framework and thus help removing the barriers of interoperability between applications and content that often prevent scientists from spending more of their precious time on solving really interesting problems. Last but not least, a significant subset of core OpenTox functionalities, developed by different partners using different programming languages and development tools/libraries, have been packaged together in a virtual appliance[1] and made available in a very convenient way for demonstration and experimental purposes. These tools are compatible with each other and rely on the OpenTox APIs and ontologies to orchestrate their work.

[1] Virtual appliance http://en.wikipedia.org/wiki/Virtual_appliance accessed on August 21st, 2011

1. Introduction

Most of the common tasks in toxicity prediction consist of several typical steps, such as access to datasets, descriptor calculation and validation procedures. Usually, the components that implement these steps are developed from scratch for every new predictive application, and this often leads to undesirable duplication of effort and/or lack of interoperability. The availability of a universal set of interoperable components could facilitate the implementation of new specialized applications that combine algorithms in the most appropriate way and allow fast and rigorous model development and testing.

The OpenTox framework consists of a set of web-accessible components that help solve common tasks in toxicity and chemical property predictions. The OECD principles of (Q)SAR validation, as well as the additional goals of software interoperability, flexibility, transparency, and extensibility, have been considered as major requirements in the design and implementation of the OpenTox framework. The design objectives were to build a component-based system, independent of programming languages and operating systems, where the components interoperate between themselves and with external software packages, which enables aggregating data from different sources and stays open for future extensions. OpenTox made two major technological choices to keep the developments within these constraints: i) the REpresentational State Transfer (REST) software architecture style[2], allowing platform and programming language independence and facilitating the implementation of new data and processing components; ii) a formally defined common information model, based on the W3C Resource Description Framework (RDF)[3] and communication through well-defined interfaces ensuring interoperability of the web components.

In the sub-sections below we discuss these principles, goals, design objectives, and technological choices and how they were incorporated into the final version of the OpenTox framework released in the FP7 project.

1.1 OECD principles of (Q)SAR validation

Here we briefly review the five OECD (Q)SAR validation principles and their relevance to the OpenTox framework development. We have incorporated these principles in the OpenTox framework design as much as possible and wherever appropriate.

1.1.1 Defined endpoint

(Q)SAR model quality crucially depends on the clarity of endpoints and experimental protocols used, and on the ability to communicate this information in an unambiguous way, both in model development and model application. The current practice usually includes a textual description of the materials and methods used for acquiring experimental data as well as literature references, while the model description is a separate entity. The challenge to the distributed web services framework described in this report was to provide an automatic and unique way of describing and linking the endpoint information in a formal way, able to be processed automatically by the software with minimal human interaction. This is currently solved by making use of a simple ontology of endpoints. We have defined an ontology, based on the OWL (Web Ontology Language)[4], for

[2] R. T. Fielding, Representational State Transfer (REST) http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, accessed on Apr 28, 2011

[3] Resource Description Framework http://www.w3.org/RDF/, accessed on Apr 28, 2011

[4] OWL Web Ontology Language Overview http://www.w3.org/TR/owl-features/, accessed on Apr 28, 2011

toxicological endpoints, which is in line with current ECHA REACH guidance[5]. Using this ontology, each attribute in a toxicological dataset can be associated with an entry to the ontology, therefore allowing a unique mapping between endpoints in various and heterogeneous datasets. This ontology possesses 5 subclasses:

- ecotoxic effects;
- environmental fate parameters;
- human health effects;
- physico-chemical effects;
- toxicokinetics.

Each of these subclasses has one or two further layers of subclasses.

The endpoint ontology can be accessed in the development documents section of the OpenTox website[6] and can be viewed with the Protégé[7] editor.

1.1.2 An unambiguous algorithm

OpenTox has defined and deployed an algorithm type ontology[8]. This ontology allows a clear definition of what type of algorithm is used to construct a model. Full description of algorithms, including references, parameters and default values has been achieved by adopting relevant classes from the Blue Obelisk ontology[9].

1.1.3 Defined applicability domain

We handle applicability domain as an algorithm or model, e.g. a specific applicability domain algorithm is applied to a dataset, and the result is then an applicability domain model. This model can then be applied to reason about the applicability of a model when applied to a novel compound.

1.1.4 Appropriate measures of goodness-of-fit, robustness and predictivity

Within the validation part of the final framework we have included validation and cross-validation objects. These objects provide access to a wide variety of measures for evaluating the quality of the models generated by applying algorithms to datasets. For classification tasks these measures include:

- Confusion matrix - a matrix where each row represents the instances in a predicted class and each column represents the instances in an actual class; one benefit of a confusion matrix is that it is easy to see if the system is confusing two or more classes;

[5] Guidance on information requirements and chemical safety assessment; Chapter R.6: QSARs and grouping of chemicals http://guidance.echa.europa.eu/docs/guidance_document/information_requirements_r6_en.pdf?vers=20_08_08, accessed on Apr 28, 2011

[6] ECHA endpoints ontology http://www.opentox.org/data/documents/development/RDF%20files/Endpoints/, accessed on Apr 28, 2011

[7] Protégé ontology editor and knowledge-base framework http://protege.stanford.edu/, accessed on Apr 28, 2011

[8] OpenTox algorithm type ontology http://www.opentox.org/data/documents/development/RDF%20files/AlgorithmTypes/view accessed on May 30, 2011

[9] Blue Obelisk ontology http://qsar.svn.sf.net/viewvc/qsar/trunk/qsar-dicts/descriptor-ontology.owl?revision=218&view=markup, accessed on Apr 28, 2011

- Absolute number and percentage of unpredicted compounds - some compounds might fall outside the applicability domain of the algorithm or model; these numbers provide an overview on the applicability domain fit for the compound set requiring prediction;
- Precision, recall, and F2-measure - these three measures give an overview on how pure and how sensitive the model is; the F2-measure combines the other two measures;
- ROC curve plot and AUC - a receiver operating characteristic (ROC) curve is a graphical plot of the true-positive rate against the false-positive rate as its discrimination threshold is varied; this gives a good understanding of how well a model is performing; as a summarizing performance scalar metric, the area under curve (AUC) is calculated from the ROC curve; a perfect model would have an area of 1.0, while a random one would have an area of 0.5.

For regressions tasks the following measures have been implemented:

- MSE and RMSE - the mean square error (MSE) and root mean squared error (RMSE) of a regression model are popular ways to quantify the difference between the predictor and the true value;
- R^2 - the explained variance (R^2) provides a measure of how well future outcomes are likely to be predicted by the model. It compares the explained variance (variance of the model's predictions) with the total variance (of the data).

1.1.5 Mechanistic interpretation, if possible

Mechanistic interpretation often relies on human knowledge and usually cannot be done automatically. However, in the final framework we generate skeletons for reporting using the validation results created by extensive testing during model construction, allowing subsequent user-entered explanations about mechanisms. QMRF reporting is facilitated by the integration of the QMRF editor into OpenTox applications allowing the end-users to annotate models with the information required by the QMRF format.

1.2 REST software architecture style

REST is a software architecture style for network-based applications, defined by Roy T. Fielding by analysing the properties of the World Wide Web (WWW) and other network architectures, and deriving the architectural constraints that made the WWW successful. There is a plethora of information on RESTful design principles[10], development frameworks and examples.

The design of a REST system, based on the HTTP protocol, starts by identifying the domain objects, followed by mapping the objects to resources and defining identifiers (Uniform Resource Identifier, URI, patterns) and operations (HTTP verbs) on each resource. In the case of OpenTox, the minimum set of domain objects, identified collaboratively by the project partners[11], consists of chemical compounds, properties of chemical compounds, datasets of chemical compounds and their properties (measured or calculated), algorithms (including descriptor calculation, regression, classification, structural alerts, quantum chemistry algorithms, etc.), predictive models (e.g., a model obtained by applying a machine learning algorithm to a training dataset), validation algorithms, and reports. In addition, tasks are introduced as special resources to allow

[10] Richardson L, Ruby S: RESTful Web Services. Sebastopol, CA, USA: O'Reilly Media, Inc; 2007

[11] Hardy B, Douglas N, Helma C, Rautenberg M, Jeliazkova N, Jeliazkov V, Nikolova I, Benigni R, Tcheremenskaia O, Kramer S, Girschick T, Buchwald F, Wicker J, Karwath A, Gütlein M, Maunz A, Sarimveis H, Melagraki G, Afantitis A, Sopasakis P, Gallagher D, Poroikov V, Filimonov D, Zakharov A, Lagunin A, Gloriozova T, Novikov S, Skvortsova N, Druzhilovsky D, Chawla S, Ghosh I, Ray S, Patel H, Escher S: Collaborative Development of Predictive Toxicology Applications. J Cheminform 2010, 2:7.

representation and handling of long running asynchronous jobs. Every resource is identified by a unique web address, following an agreed pattern specific to the resource type (e.g., /algorithm/{id} for algorithms, /compound/{id} for compounds, etc.). The resources can be created (HTTP POST), updated (HTTP PUT) and deleted (HTTP DELETE), or their representations retrieved (HTTP GET).

1.3 Common information model

The common information model of the OpenTox domain objects is based on the Resource Description Framework (RDF) and described by the OpenTox ontology[12]. It should be noted that the initial design of the OpenTox API (version 1.0)[13] was based on an XML schema, but it was later decided to adopt RDF as a more powerful approach to describe objects and their relationships, as well as to facilitate the reuse of ongoing ontology developments in bioinformatics.

RDF allows defining a data model (how the data is organized), instead of specifying a data format (how the data is written into a file). Once a data model is defined, it could be serialized into different formats, for example RDF/XML[14], N3[15], or TURTLE[16]. The OWL Web Ontology Language is built on top of RDF, and, compared to RDF, imposes restrictions on what is allowed to be represented. Because of such restrictions, the OWL subsets OWL-Lite and OWL-DL (Description Logic) allow performing automated machine reasoning.

1.4 Combining REST and RDF

Both REST and RDF technologies encourage data model development and consider assigning resource identifiers important. However, there are differences, as REST identifiers are used as addresses of the underlying protocol (e.g. HTTP URIs) and it is essential that URIs are dereferenceable[17]. While the RDF representation allows HTTP URIs as resource identifiers, these are considered names, not addresses, and are not necessarily dereferenceable. HTTP URIs are hierarchical, while RDF does not exploit the hierarchy and splits HTTP URIs into a prefix and an identifier instead. REST resources define clear boundaries between information entities, while data, represented via RDF, is usually perceived as one linked graph. The common usage of RDF for data integration is to convert data coming from diverse sources into a (typically read only) single triple storage[18] and provide a query interface (SPARQL Protocol and RDF Query Language, SPARQL, endpoint). On the contrary, web services provide distributed and dynamically generated information. Most REST services define data formats[19] as a means for communication, rather than an explicit data model. The simultaneous use of RDF and REST is not yet widespread and there are ongoing debates on various related topics. Nevertheless, there is an added value of merging both technologies for independent deployments of multiple services, able to dynamically generate linked data with dereferenceable links. This could lead to an enrichment of the information space and scalability, in a manner similar to a deployment of many web servers that provide hypertext documents.

[12] OpenTox ontology http://opentox.org/api/1.1/opentox.owl, accessed on Apr 29, 2011

[13] OpenTox API v1.0 (archived version) http://www.opentox.org/dev/apis/api-1.0, accessed on Apr 29, 2011

[14] RDF/XML Syntax Specification http://www.w3.org/TR/REC-rdf-syntax/, accessed on Apr 29, 2011

[15] Notation 3 (N3) A Readable RDF Syntax http://www.w3.org/DesignIssues/Notation3, accessed on Apr 29, 2011

[16] Turtle - Terse RDF Triple Language http://www.w3.org/TeamSubmission/turtle/, accessed on Apr 29, 2011

[17] Dereferenceable URI, http://en.wikipedia.org/wiki/Dereferenceable_Uniform_Resource_Identifier, accessed on Sept 15, 2011

[18] Wikipedia Entry for Triplestore, http://en.wikipedia.org/wiki/Triplestore, accessed on Apr 30, 2011

[19] Microformats http://microformats.org/, accessed on Apr 29, 2011

The OpenTox framework integrates both technologies into a distributed web services framework, where both data and processing resources are described by ontologies: either existing ones or developed within the project. The framework consists of modules, developed by different partners and with different programming languages, running on a set of geographically dispersed servers and communicating via the Internet. The modules can be used to build more complex applications, to embed OpenTox web services into workflows, to build web mashups, to consume web services via rich client applications, and can additionally be collapsed into a local application.

2. OpenTox design principles

The principles of interoperability, flexibility, transparency and extensibility are key ingredients of the OpenTox framework design, guiding its architecture and implementation.

2.1 Interoperability

Interoperability with respect to the OpenTox Framework refers to the principle that different OpenTox components or services may correctly exchange information with each other and subsequently make use of that information. Both syntactic interoperability for correct data exchange and semantic interoperability supporting the accurate communication of meaning and interpretation of data are supported principles for OpenTox resources. The principles are reflected design-wise in the use of open, standardised interfaces and ontologies. The principles are relevant in application development and deployment when a combination of distributed multiple services can provide value to a user in completing a use case satisfactorily.

2.2 Flexibility

As there exists a significant variety of user scenarios, requirements and use cases in predictive toxicology, flexibility is a key principle incorporated into OpenTox. Through the use of a component-based approach and the incorporation of the interoperability principles, many different and customised applications can be assembled that are based on the underlying platform.

2.3 Transparency

To achieve the scientific objective of knowledge-based enquiry based on principles of reasoning, reproducibility, and reliability, OpenTox supports the principle of transparency in its design. Computational models should be available for scrutiny by other scientists in as complete a manner and detail as possible. Evaluators and regulators should be able to both understand the details and accurately reproduce the results of predictive toxicity models, and be able to reliably form judgements on their validity as evidence. The principle also supports achievement of the OECD validation principles such as an unambiguous algorithm and a mechanistic interpretation, if possible. The use of open source, open interfaces and standards, and open data within the OpenTox framework supports the implementation of this transparency principle in the context of *in silico*-based predictive toxicology applications and their reported results.

2.4 Extensibility

The field of predictive toxicology is rapidly developing and broadening in many areas including the use of biomarkers, systems biology, epigenetics, toxicokinetics, *in vitro* assays, stem cell technology, computational biology etc. Hence, OpenTox needs to be extensible to a broad range of future predictive toxicology applications. In such applications, contributing and diverse experimental data and models need to be combined as evidence supporting integrated testing, safety assessment and regulatory reporting as stipulated under REACH. As the OpenTox framework has been designed with these requirements in mind, it not only provides a generic solution for (Q)SAR model development and application, but is also applicable to additional areas of scientific enquiry in the predictive toxicology field and could be extended further as part of its evolutionary development beyond the end of the OpenTox FP7 project.

3. OpenTox implementation principles

OpenTox is an open source project and tries to follow the best practices of open source project management. This means that source code, technical discussions and documents are open to the general public and interested parties can participate in development. Within the design of the framework we have concentrated on a number of principles: open source, open and distributed access, open interfaces and standards, and open data.

In the following sections we describe these implementation principles in more detail.

3.1 Open source

As the open source philosophy is inherently important for this project, all tools developed are openly available via public repositories. Further details and up-to-date links to the source code are provided at the OpenTox web site[20]. Within the framework, a variety of programming and scripting languages have been employed, such as Java, Ruby, C++, Delphi, JavaScript and PHP. This has helped ensure the programming language independence of the OpenTox API.

3.2 Open and distributed access

As it was already mentioned in the introductory section of this report, the OpenTox web services adhere to the REST web service architecture for sharing data and functionality among loosely-coupled, heterogeneous systems. The REST architecture is based on five key principles:

- every resource can be uniquely identified;
- use of standard HTTP;
- multiple representations of resources are allowed;
- use of hypertext links for linking of resources;
- communication is stateless.

Adhering to these principles, the REST web service architecture has a number of desired advantages when compared to other web service architectures:

- it is lightweight, as only some additional XML mark-up is required;
- the produced results are human-readable, i.e. the resources are uniquely identified by URIs and described by representations;
- RESTful web services are typically stateless and scalable;
- the produced web services have a uniform interface (the only allowed operations are the HTTP operations);
- components manipulate resources by exchanging representations of the resources.

The choice of employing web services allows the complete framework to operate in different locations, independent of operating systems and underlying implementation details.

[20] OpenTox Downloads page http://www.opentox.org/downloads/, accessed on Apr 30, 2011

3.3 Open interfaces and standards

The publicly available OpenTox application programming interface (API)[21] allows the cheminformatics and bioinformatics communities to participate in the further development of the OpenTox framework. It is based on open standards such as HTTP, REST and RDF, which turn out to provide a very solid base for building a scalable distributed framework of web services, operating with and offering access to Linked Data[22]. A clear advantage of the availability of RDF representations for the OpenTox objects data, algorithms, and models is that it allows to combine easily the RDF representations of remote resources into a standard triplestore[23], annotating and cross-linking objects with terms from existing ontologies. Publishing a dataset of chemical structures and their properties as linked data becomes as straightforward as uploading a *sdf* file to an OpenTox dataset service, with optional subsequent annotation of property tags.

3.4 Open data

Open data[24] is a philosophy and practice requiring that certain data be freely available to everyone, without restrictions from copyright, patents or other mechanisms of control. The OpenTox framework provides consistent means for stating the conditions of data ownership, licensing and re-use. This helps addressing cases where the lack of an explicit license makes it difficult to determine the status of a data set and might unintentionally restrict the use of data offered in an open spirit. In addition, when uploading data in RDF format, the client has full control of the data fields' representation. This is a substantial improvement over most of the current practices with popular chemical formats, which usually involve describing the meaning of the fields in separate documents, targeted at human readers; sadly, this approach tends to lead to quite frequent peculiarities.

4. Final framework implementation

To ensure reliable interoperability between the various web services, a well-defined API is required. The API specifies how each OpenTox web service can be used, and how the returned resources look like. It further specifies the HTML status codes returned in case of succeeded operations as well as errors codes. The OpenTox API has been documented in the OpenTox report D1.3 (Prototype evaluation, improved API and interface designs)[25]. The current version of the OpenTox API with all the latest updates and new features is available at the OpenTox website[26]. This section presents the OpenTox web services, applications, workflows and client libraries that currently implement the OpenTox API and are deployed at different locations around the world and/or available for download. A total of 27 web services, 11 applications and 4 client libraries have been implemented and deployed by 7 OpenTox partners and one third-party. The following matrix summarizes the existing independent implementations and provides pointers to the deployed services and applications. The first column of the matrix lists the sections of the OpenTox API that are implemented by the corresponding web services developed by project partners. The two rows at the bottom of the matrix list the available end user applications, workflows and client libraries, again grouped by their corresponding developers.

[21] OpenTox API v1.2 http://www.opentox.org/dev/apis/api-1.2, accessed on Apr 30, 2011

[22] Wikipedia entry for Linked Data http://en.wikipedia.org/wiki/Linked_Data, accessed on Apr 30, 2011

[23] Wikipedia entry for Triplestore http://en.wikipedia.org/wiki/Triple_store, accessed on Apr 30, 2011

[24] Wikipedia entry for Open Science Data http://en.wikipedia.org/wiki/Open_science_data, accessed on Apr 30, 2011

[25] OpenTox deliverable D1.3 (Prototype evaluation, improved API and interface designs) http://www.opentox.org/data/documents/development/opentoxreports/opentoxreportd13/, accessed on Apr 30, 2011

[26] OpenTox APIs http://www.opentox.org/dev/apis, accessed on Apr 30, 2011

OpenTox API Implementation Matrix

(list of existing independent implementations and pointers to deployed services and applications)

	ALU-FR	IBMC	IDEA	IST	NTUA	SL-JNU	TUM	3rd parties
Compound			X	X				
Dataset			X	X				
Feature			X					
Algorithm		X	X	X	X	X	X	
Model		X	X	X	X	X	X	
Ontology			X					
Validation & Reporting	X							
Task		X	X	X	X	X	X	
AA & Policy Management				OpenAM X				
End-User Apps & Workflows	QMRF Editor CheS-Mapper	IBMCDesc	ToxPredict	ToxCreate lazar	Q-Edit	MaxToxMCSS	ToxDesc Taverna	Bioclipse Bioclipse
Client Libraries			OT-client OT-sso	opentox-ruby	ToxOtis			

Last update: October 13, 2011

In the following subsections we provide some further details on these web services, end-user applications, workflows and client libraries.

4.1 OpenTox compound web services

The OpenTox compound API[27] provides different representations for chemical compounds with a unique and defined chemical structure. This section presents two OpenTox compound web services developed by IDEA and IST.

[27] OpenTox compound API http://www.opentox.org/dev/apis/api-1.2/structure, accessed on Apr 30, 2011

4.1.1 OpenTox compound web service implemented by IDEA

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/viewvc/ambit/trunk/ambit2-all/ambit2-www/
- Binaries: http://ambit.sourceforge.net/downloads.html
- Deployment requirements: Apache Tomcat[28] 6.0 or newer, MySQL[29] 5.1 or newer
- Live demo: http://apps.ideaconsult.net:8080/ambit2/compound/

4.1.2 OpenTox compound web service implemented by IST

- Programming language: Ruby
- Source code repository: http://github.com/opentox/compound/
- Deployment requirements: Debian[30]-based Linux distribution
- Installer: http://github.com/opentox/install/
- Live demo: http://webservices.in-silico.ch/compound/

4.2 OpenTox dataset web services

The OpenTox dataset API[31] provides access to chemical compounds and their features (e.g. structural, physical-chemical, biological, toxicological properties). This section presents two OpenTox dataset web services developed by IDEA and IST.

4.2.1 OpenTox dataset web service implemented by IDEA

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/viewvc/ambit/trunk/ambit2-all/ambit2-www/
- Binaries: http://ambit.sourceforge.net/downloads.html
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Live demo: http://apps.ideaconsult.net:8080/ambit2/dataset/

4.2.2 OpenTox dataset web service implemented by IST

- Programming language: Ruby
- Source code repository: http://github.com/opentox/dataset/
- Deployment requirements: Debian-based Linux distribution
- Installer: http://github.com/opentox/install/

[28] Apache Tomcat servlet container http://tomcat.apache.org/, accessed on Apr 30, 2011

[29] MySQL Relational Database Management System http://www.mysql.com/, accessed on Apr 30, 2011

[30] http://www.debian.org/ accessed on May 31, 2011

[31] OpenTox dataset API http://www.opentox.org/dev/apis/api-1.2/dataset, accessed on Apr 30, 2011

- Live demo: http://webservices.in-silico.ch/dataset/

4.3 OpenTox feature web services

The OpenTox feature API[32] provides access to objects representing any kind of property, assigned to a Compound. The feature types are determined via their links to ontologies (Feature ontologies, Descriptor ontologies, Endpoints ontologies). This section presents an OpenTox feature web service developed by IDEA.

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/viewvc/ambit/trunk/ambit2-all/ambit2-www/
- Binaries: http://ambit.sourceforge.net/downloads.html
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Live demo: http://apps.ideaconsult.net:8080/ambit2/feature/

4.4 OpenTox algorithm web services

The OpenTox algorithm API[33] provides access to OpenTox algorithms. This section presents six OpenTox algorithm web services developed by IBMC, IDEA, IST, NTUA, SL-JNU, and TUM.

4.4.1 OpenTox algorithm web service implemented by IBMC

- Programming language: Delphi, PHP
- Source code: http://ibmc.svn.sourceforge.net/viewvc/ibmc/OpenTox
- Installer: http://ibmc.svn.sourceforge.net/viewvc/ibmc/install/OpenTox_IBMC.rar
- Deployment requirements: Apache 2.63 or newer, PHP 5.0 or newer, MySQL 5.33 or newer
- Live demo: http://195.178.207.160/OpenTox/Algorithm

4.4.2 OpenTox algorithm web service implemented by IDEA

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/viewvc/ambit/trunk/ambit2-all/ambit2-www/
- Binaries: http://ambit.sourceforge.net/downloads.html
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Live demo: http://apps.ideaconsult.net:8080/ambit2/algorithm/

4.4.3 OpenTox algorithm web service implemented by IST

- Programming language: Ruby
- Source code repository: http://github.com/opentox/algorithm/
- Installer: http://github.com/opentox/install/

[32] OpenTox feature API http://www.opentox.org/dev/apis/api-1.2/Feature, accessed on Apr 30, 2011

[33] OpenTox algorithm API http://www.opentox.org/dev/apis/api-1.2/Algorithm, accessed on Apr 30, 2011

- Deployment requirements: Debian-based Linux distribution
- Live demo: http://webservices.in-silico.ch/algorithm (REST interface)

4.4.4 OpenTox algorithm web service implemented by NTUA

- Programming language: Java
- Source code repository: https://github.com/alphaville/jaqpot3
- Installation and dependencies: further details are provided in the online tutorial available at http://opentox.ntua.gr/index.php/blog/14-editors?showall=1
- Deployed instance: http://opentox.ntua.gr:8080/ (get a list of available algorithm web services at http://opentox.ntua.gr:8080/algorithm)

4.4.5 OpenTox algorithm web service implemented by SL-JNU

- Programming language: Java
- Source code: http://opentox2.informatik.uni-freiburg.de/files/MaxtoxMCSS.src.zip
- Installer: http://opentox2.informatik.uni-freiburg.de/files/MaxtoxMCSS.war
- Dependencies: cdk-1.3.4[34], restlet-2.1[35], jena-2.6.2[36], jchempaint-8[37], JNI_SVM-light-6.01[38], Rserve[39]
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Available algorithms: http://opentox2.informatik.uni-freiburg.de:8080/MaxtoxMCSS/algorithm

4.4.6 OpenTox algorithm web service implemented by TUM

- Programming language: Java
- Source code repository: http://opentox.informatik.tu-muenchen.de/trac/TUMOpenTox/browser/trunk/src/tum
- Binaries: http://opentox.informatik.tu-muenchen.de/trac/TUMOpenTox/wiki/Download
- Deployment requirements: Apache Tomcat 6.0 or newer, PostgreSQL[40] 8.4.7 or newer
- Available algorithms: http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm

4.5 OpenTox model web services

The OpenTox model API[41] provides access to OpenTox models. This section presents six OpenTox model web services developed by IBMC, IDEA, IST, NTUA, SL-JNU, and TUM.

[34] The Chemistry Development Kit, http://cdk.sourceforge.net/ accessed on June 7, 2011

[35] Restlet RESTful web framework for Java, http://www.restlet.org/ accessed on June 7, 2011

[36] Jena Semantic Web Framework, http://jena.sourceforge.net/ accessed on June 7, 2011

[37] JChemPaint, http://jchempaint.sourceforge.net/ accessed on June 7, 2011

[38] JNI_SVM-light-6.01, http://www.mpi-inf.mpg.de/~mtb/svmlight/JNI_SVM-light-6.01.zip accessed on June 7, 2011

[39] Rserve, http://www.rforge.net/Rserve/ accessed on June 7, 2011

[40] PostgreSQL http://www.postgresql.org/, accessed on Apr 30, 2011

[41] OpenTox model API http://www.opentox.org/dev/apis/api-1.2/Model, accessed on Apr 30, 2011

4.5.1 OpenTox model web service implemented by IBMC

- Programming language: Delphi, PHP
- Source code: http://ibmc.svn.sourceforge.net/viewvc/ibmc/OpenTox
- Installer: http://ibmc.svn.sourceforge.net/viewvc/ibmc/install/OpenTox_IBMC.rar
- Deployment requirements: Apache 2.63[42] or newer, PHP 5.0[43] or newer, MySQL 5.33 or newer
- Live demo: http://195.178.207.160/OpenTox/model

4.5.2 OpenTox model web service implemented by IDEA

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/viewvc/ambit/trunk/ambit2-all/ambit2-www/
- Binaries: http://ambit.sourceforge.net/downloads.html
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Live demo: http://apps.ideaconsult.net:8080/ambit2/model/

4.5.3 OpenTox model web service implemented by IST

- Programming language: Ruby
- Source code repository: http://github.com/opentox/model/
- Installer: http://github.com/opentox/install/
- Deployment requirements: Debian-based Linux distribution
- Live demo: http://webservices.in-silico.ch/model/ (REST interface)

4.5.4 OpenTox model web service implemented by NTUA

- Programming language: Java
- Source code repository: https://github.com/alphaville/jaqpot3
- Installation and dependencies: further details are provided in the online tutorial available at http://opentox.ntua.gr/index.php/blog/14-editors?showall=1
- Deployed instance: http://opentox.ntua.gr:8080/ (get a list of all models available on the server at http://opentox.ntua.gr:8080/model; you can search for model trained with some particular algorithm using the URI parameter *algorithm* (e.g. http://opentox.ntua.gr:8080/model?algorithm=fastRbfNn will return a list of all RBF-NN models)

4.5.5 OpenTox model web service implemented by SL-JNU

- Programming language: Java
- Source code: http://opentox2.informatik.uni-freiburg.de/files/MaxtoxMCSS.src.zip

[42] Apache HTTP server http://httpd.apache.org/ accessed on June 9, 2011

[43] PHP hypertext pre-processor http://www.php.net/ accessed on June 9, 2011

- Installer: http://opentox2.informatik.uni-freiburg.de/files/MaxtoxMCSS.war
- Dependencies: cdk-1.3.4, restlet-2.1, jena-2.6.2, jchempaint-8, JNI_SVM-light-6.01, Rserve
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Available models: http://opentox2.informatik.uni-freiburg.de:8080/MaxtoxMCSS/model

4.5.6 OpenTox model web service implemented by TUM

- Programming language: Java
- Source code repository: http://opentox.informatik.tu-muenchen.de/trac/TUMOpenTox/browser/trunk/src/tum
- Binaries: http://opentox.informatik.tu-muenchen.de/trac/TUMOpenTox/wiki/Download
- Deployment requirements: Apache Tomcat 6.0 or newer, PostgreSQL 8.4.7 or newer
- Available models: http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/model

4.6 OpenTox ontology web services

The OpenTox ontology API[44] provides storage and search functionality for objects defined in OpenTox services and relevant ontologies. This section presents an OpenTox ontology web service developed by IDEA.

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/svnroot/ambit/branches/opentox/ontology-service
- Binaries: http://ambit.sourceforge.net/downloads.html
- Deployment requirements: Apache Tomcat 6.0 or newer
- Live demo: http://apps.ideaconsult.net:8080/ambit2/ontology/

4.7 OpenTox validation and reporting web services

The OpenTox validation API[45] provides access to OpenTox validation procedures. This section presents an OpenTox validation web service developed by ALU-FR.

- Programming language: Ruby
- Source code repository: http://github.com/opentox/validation
- Binaries: http://www.maunz.de/opentox/Opentox%20IST%20Virtual%20Appliance.ova[46]
- Deployment requirements: Debian based operating system, see http://github.com/opentox/install for installation instructions

[44] OpenTox ontology API http://www.opentox.org/dev/apis/api-1.2/Ontology, accessed on Apr 30, 2011

[45] OpenTox validation API http://www.opentox.org/dev/apis/api-1.2/Validation, accessed on Apr 30, 2011

[46] See http://www.maunz.de/wordpress/opentox/2011/using-ist-opentox-virtual-appliance-under-windows for installation instructions

4.8 OpenTox task web services

The OpenTox task API[47] provides means for handling asynchronous jobs via an intermediate task resource. This section presents six OpenTox task web services implemented by IBMC, IDEA, IST, NTUA, SL-JNU, and TUM.

4.8.1 OpenTox task web service implemented by IBMC

- Programming language: Delphi, PHP
- Source code: http://ibmc.svn.sourceforge.net/viewvc/ibmc/OpenTox
- Installer: http://ibmc.svn.sourceforge.net/viewvc/ibmc/install/OpenTox_IBMC.rar
- Deployment requirements: Apache 2.63 or newer, PHP 5.0 or newer, MySQL 5.33 or newer
- Live demo: http://195.178.207.160/OpenTox/task

4.8.2 OpenTox task web service implemented by IDEA

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/viewvc/ambit/trunk/ambit2-all/ambit2-www/
- Binaries: http://ambit.sourceforge.net/downloads.html
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Live demo: http://apps.ideaconsult.net:8080/ambit2/task/

4.8.3 OpenTox task web service implemented by IST

- Programming language: Ruby
- Source code repository: http://github.com/opentox/task/
- Installer: http://github.com/opentox/install/
- Deployment requirements: Debian-based Linux distribution
- Live demo: http://webservices.in-silico.ch/task/ (REST interface)

4.8.4 OpenTox task web service implemented by NTUA

- Programming language: Java
- Source code repository: https://github.com/alphaville/jaqpot3
- Installation and dependencies: further details are provided in the online tutorial available at http://opentox.ntua.gr/index.php/blog/14-editors?showall=1
- Deployed instance: http://opentox.ntua.gr:8080/ (get a list of all tasks at http://opentox.ntua.gr:8080/task; tasks can be retrieved by status using the URI parameter *status*. (e.g. http://opentox.ntua.gr:8080/task?status=cancelled will return all cancelled tasks)

4.8.5 OpenTox task web service implemented by SL-JNU

- Programming language: Java

[47] OpenTox task API http://www.opentox.org/dev/apis/api-1.2/AsyncTask, accessed on Apr 30, 2011

- Source code: http://opentox2.informatik.uni-freiburg.de/files/MaxtoxMCSS.src.zip
- Installer: http://opentox2.informatik.uni-freiburg.de/files/MaxtoxMCSS.war
- Dependencies : cdk-1.3.4, restlet-2.1, jena-2.6.2, jchempaint-8, JNI_SVM-light-6.01, Rserve
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Live demo: http://opentox2.informatik.uni-freiburg.de:8080/MaxtoxMCSS/task

4.8.6 OpenTox task web service implemented by TUM

- Programming language: Java
- Source code repository: http://opentox.informatik.tu-muenchen.de/trac/TUMOpenTox/browser/trunk/src/tum
- Binaries: http://opentox.informatik.tu-muenchen.de/trac/TUMOpenTox/wiki/Download
- Deployment requirements: Apache Tomcat 6.0 or newer, PostgreSQL 8.4.7 or newer
- Online demo: http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/task

4.9 OpenTox authentication, authorisation and policy management web services

The OpenTox authentication, authorisation and policy management API[48] handles granting access to protected resources to authorised users and provides means for managing resource access policies. This section presents OpenTox authentication, authorisation and policy management web services developed and deployed by IST.

We have considered different authentication and authorisation (A&A) solutions and have chosen OpenSSO, a single sign-on authentication and authorisation server initially developed by Sun Microsystems[49]. A policy configuration service has also been developed to define and manage the access control policies. Finally, a common user and policy database has been established. More details on the required dependencies, the installation procedures, pointers to the source code, as well as example queries are available at http://www.opentox.org/dev/documentation/Guidance%20on%20Using%20AA#set-up-a-a

Because the OpenTox API is based on the REST architecture, implementing a complete A&A solution is not a trivial task. While use of REST is becoming increasingly popular over the Internet, the security technologies to support it are not well established yet or a subject of current research. For this reason, the solution presented in this deliverable is considered merely an initial step to investigate the problem and gain important experience. It is aimed to be a current working and usable solution, but there is certainly scope for future extensions, including the implementation of entirely new concepts.

4.9.1 OpenSSO/OpenAM

OpenSSO is a single sign-on authentication and authorisation server developed by Sun Microsystems, which would, however, require a custom REST policy web service due to currently missing support. REST interfaces are of increasing importance, as exemplified in the following quote from the OpenSSO developers (1):

> *“The recent rapid advancements and adoption of Web services, service-oriented architecture (SOA), and Representational State Transfer (REST) architecture within enterprises have left the industry wanting more. Organizations and developers, such as those who focus on Web 2.0, are demanding interface support from*

[48] OpenTox authentication, authorisation and policy management API http://www.opentox.org/dev/apis/api-1.2/AA, accessed on Apr 30, 2011

[49] On January 27, 2010, Sun was acquired by Oracle Corporation and was subsequently renamed Oracle America, Inc.

> *identity and access management software. The Open Web SSO Project, called OpenSSO for short, answers those demands."*

OpenAM is a community-based clone of OpenSSO. OpenAM maintains 100% compatibility to existing OpenSSO versions[50]. It is maintained by ForgeRock, a Norwegian software company with a focus on open source enterprise software[51]. There is also a wiki available[52]. The OpenSSO Users Mailinglist[53] is very active and provides contact to experts in the field of authentication and authorisation.

4.9.2 OpenLDAP

OpenLDAP is an LDAPv3-compatible directory server intended to hold identity information. OpenSSO can attach a variety of such databases in a very flexible manner, including Microsoft's Active Directory, and generic LDAP servers. We use OpenLDAP as a common backend to both the PLONE Content Management System (CMS), on which the opentox.org site is currently based, and to OpenSSO.

The article series *Securing Applications With Identity Services*[54] describes in detail the setup of an OpenSSO deployment, including the client application. Installation of the server is straightforward - it is only necessary to extract and start the web server and to deploy OpenSSO as a web application (Web application Archive, WAR, file[55]) within it. Any J2EE (Java 2 Platform, Enterprise Edition) compatible web server can be used (e.g. Tomcat, Glassfish).

4.9.3 General Overview

Securing Applications With Identity Services describes comprehensively the REST interface of OpenSSO. The full interface is available within OpenTox, while this documentation only describes a subset of it. In particular, one will also find services for checking validity of tokens ("Token validation"), for finding which user a specific token belongs to ("Attributes"), or for log out.

How does Authentication and Authorisation work with OpenSSO? In a nutshell, the user authenticates against the OpenSSO server at the URI http://<hostname>:<port>/auth/authenticate by POSTing "username" and "password", and in turn receives a token. The token may be used to get authorisation from the OpenSSO server to access (i.e. GET, POST, PUT, DELETE) a specific URI. The process of deciding authorisation is governed by policies stored on the server. The authorisation request itself consists of a POST to http://<hostname>:<port>/auth/authorize. The request should contain the target URI, one of the four access methods and a valid token. In case of a grant, a Boolean value *true* is returned as content. In case of a deny, Boolean *false* is returned.

[50] Sun Microsystems has been acquired by Oracle Inc., who, in turn, have decided to discontinue OpenSSO

[51] ForgeRock Website (Downloads). http://forgerock.com/downloads.html accessed on May 30, 2011

[52] ForgeRock. *OpenAM WIKI.* https://wikis.forgerock.org/confluence/display/openam/Home accessed on May 30, 2011

[53] OpenSSO user's mailing list. http://java.net/projects/opensso/lists/users/archive accessed on May 30, 2011

[54] Ranganathan, Aravindan. Securing Applications With Identity Services. http://developers.sun.com/identity/reference/techart/id-svcs.html accessed on May 30, 2011

[55] Wikipedia article on WAR file format (Sun), http://en.wikipedia.org/wiki/WAR_%28Sun_file_format%29, accessed on Sept 15, 2011

A lot of backends (e.g. LDAP servers) can be attached to OpenSSO to validate subject identity data. Also, very flexible policies can be created using wildcards and overlapping policies. The flow diagram in Figure 1 describes an example use case involving authentication and authorisation based on OpenSSO.

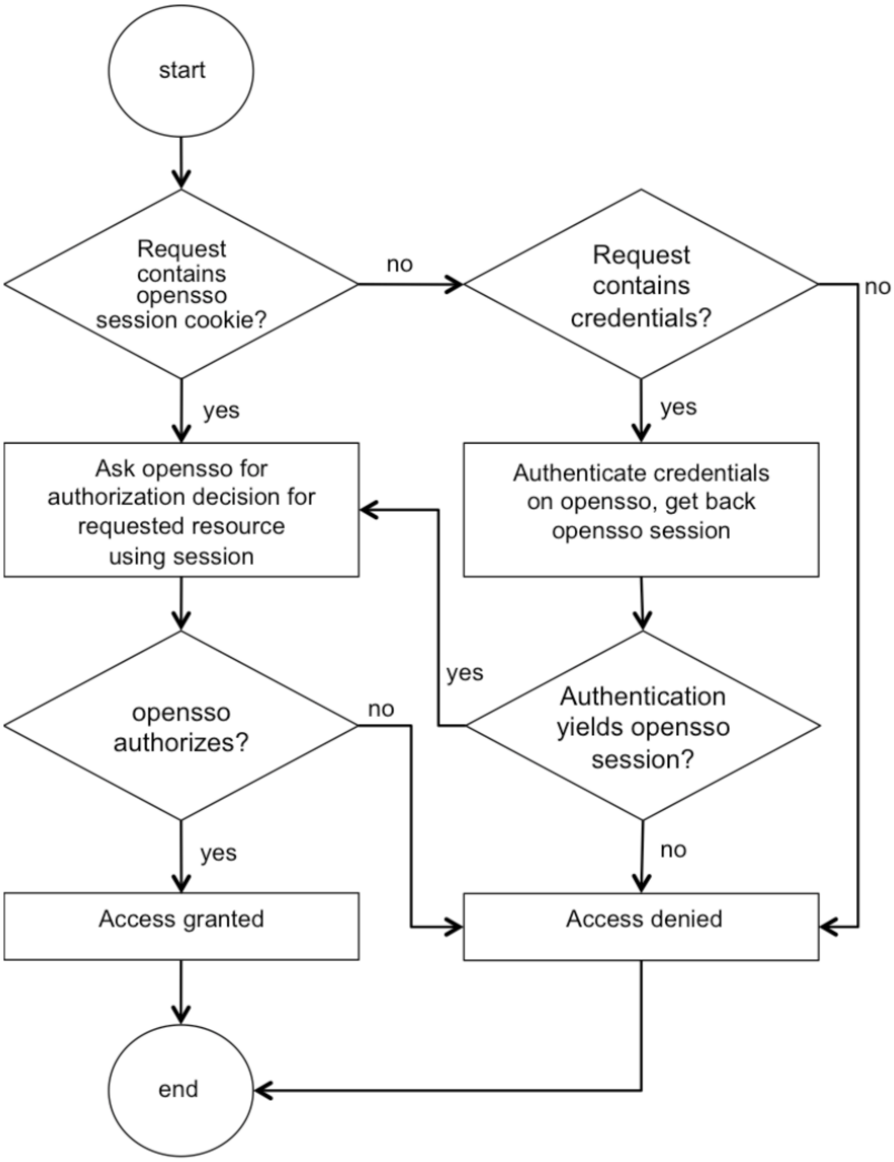

start
Request contains opensso session cookie?
no
Request contains credentials?
no
yes
yes
Ask opensso for authorization decision for requested resource using session
Authenticate credentials on opensso, get back opensso session
yes
opensso authorizes?
no
Authentication yields opensso session?
yes
no
Access granted
Access denied
end

Figure 1 Authentication and Authorisation flow diagram

4.9.4 Security

All traffic circulating user IDs, and passwords should be SSL-secured for privacy.

Since SSL is a standard technology for the web, server certificates should be easy to install and maintain. They are also available free of charge[56].

[56] StartSSL http://www.startssl.com/ accessed on May 31, 2011

4.10 End user applications and workflows

End user applications and workflows consume the OpenTox web services and provide friendly graphical user interfaces to the functions provided by these web services. This section presents eleven end-user applications and workflows developed by ALU-FR, IBMC, IDEA, IST, NTUA, SL-JNU, TUM and one 3rd party.

4.10.1 QMRF editor application implemented by ALU-FR

This is an extension to the existing QMRF[57] Editor with the ability to employ web services as well as authorization and authentication.

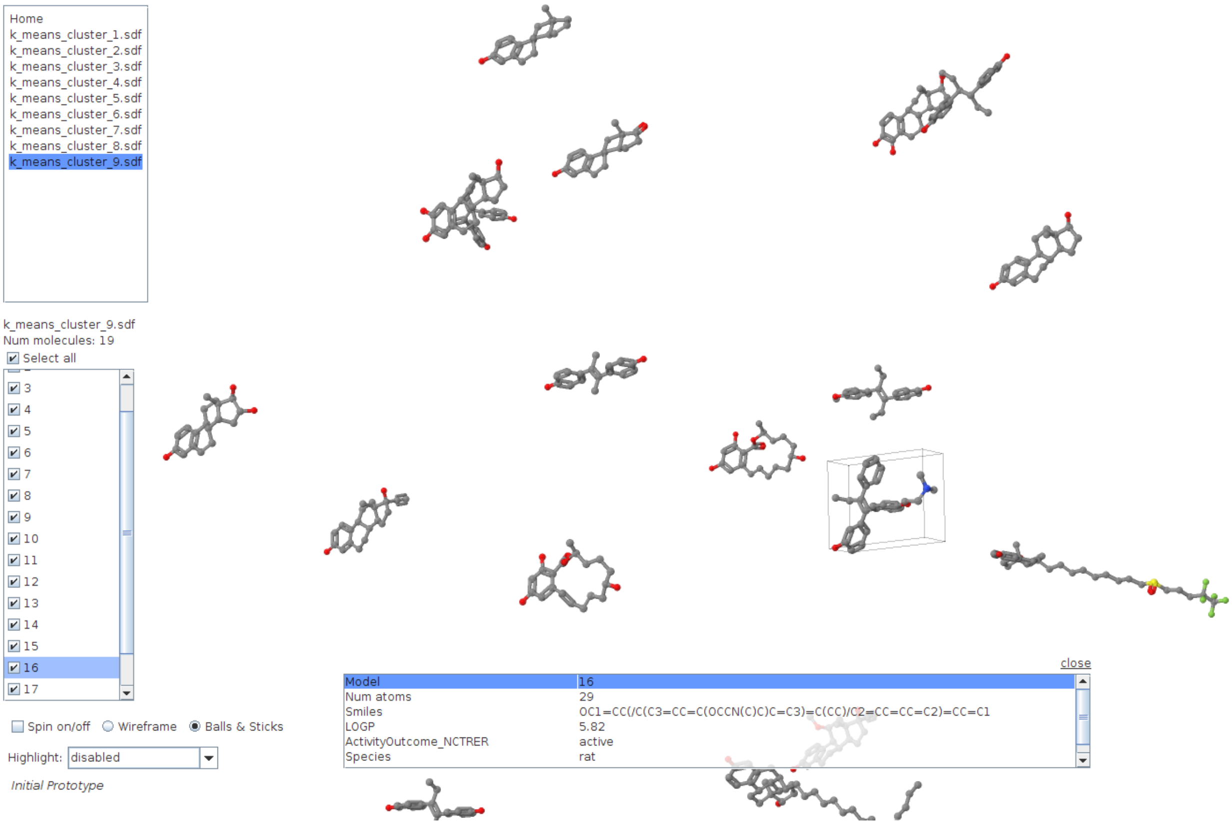
- Programming language: Java
- Source code repository: https://github.com/mguetlein/qmrfedit
- Binaries: http://opentox.informatik.uni-freiburg.de/qmrfedit/
- Live demo: http://opentox.informatik.uni-freiburg.de/qmrfedit/OT_QMRFEditor.jnlp (Java web start application)

[57] QSAR Model Reporting Format Inventory http://qsardb.jrc.it/qmrf/ accessed on August 21, 2011

4.10.2 CheS-Mapper application implemented by ALU-FR

The CheS-Mapper application was presented at VIZBI 2011, Workshop on Visualizing Biological Data, 16-19 March 2011, Broad Institute, Cambridge-MA, USA, see http://vizbi.org/2011/Posters/Collection/?poster=C09 for some further details.

- Programming language: Java
- Source code repository: http://github.com/mguetlein/ches-mapper
- Binaries: http://ortona.informatik.uni-freiburg.de/ches-mapper



Home
k_means_cluster_1.sdf
k_means_cluster_2.sdf
k_means_cluster_3.sdf
k_means_cluster_4.sdf
k_means_cluster_5.sdf
k_means_cluster_6.sdf
k_means_cluster_7.sdf
k_means_cluster_8.sdf
k_means_cluster_9.sdf

k_means_cluster_9.sdf
Num molecules: 19
☑ Select all

☑ 3
☑ 4
☑ 5
☑ 6
☑ 7
☑ 8
☑ 9
☑ 10
☑ 11
☑ 12
☑ 13
☑ 14
☑ 15
☑ 16
☑ 17

☐ Spin on/off ○ Wireframe ● Balls & Sticks

Highlight: disabled

Initial Prototype

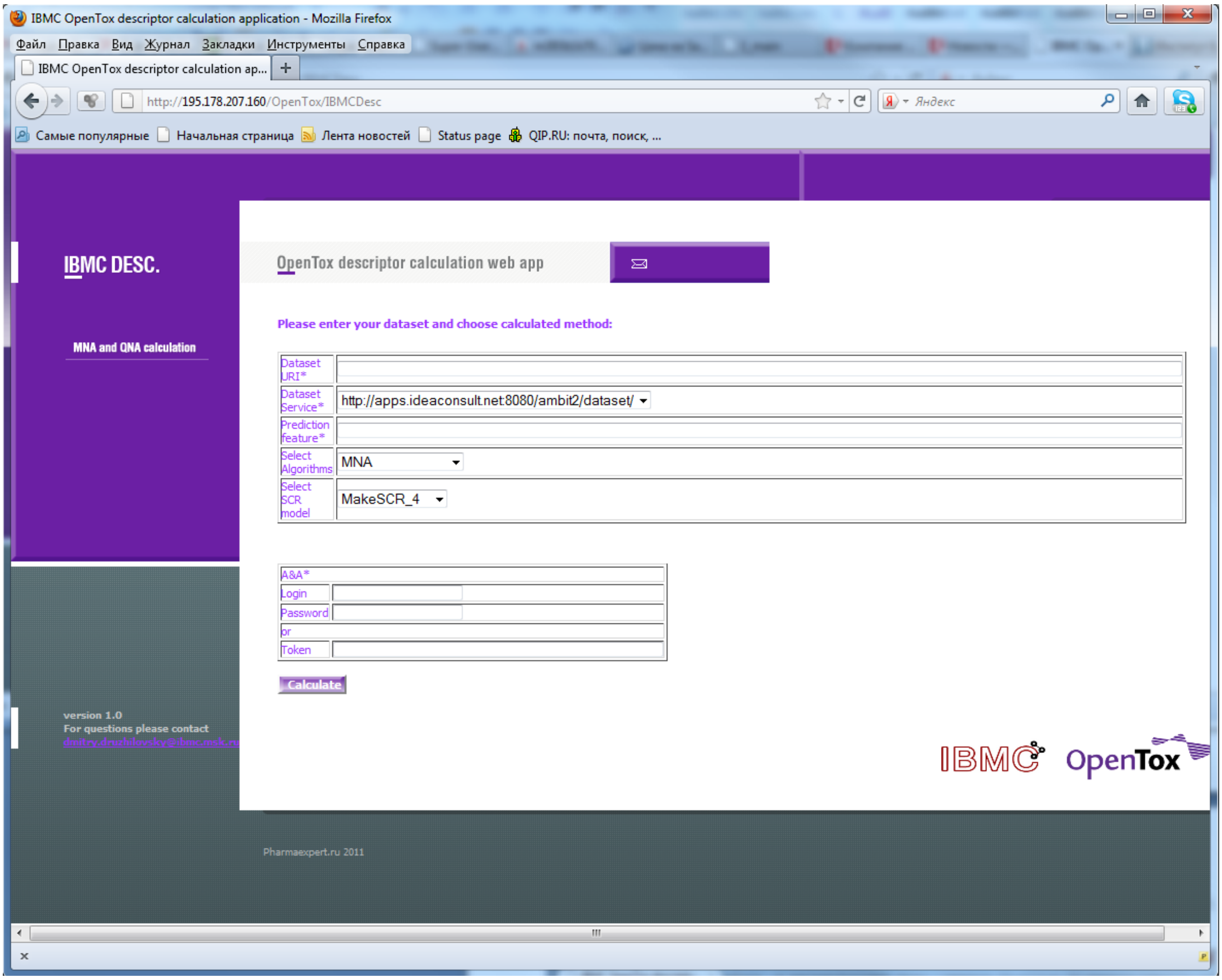
close

Model	16
Num atoms	29
Smiles	OC1=CC(/C(C3=CC=C(OCCN(C)C)C=C3)=C(CC)/C2=CC=CC=C2)=CC=C1
LOGP	5.82
ActivityOutcome_NCTRER	active
Species	rat

Figure 2 Screenshot of CheS-Mapper application

4.10.3 IBMCDesc application implemented by IBMC

- Programming language: Delphi, PHP
- Source code: http://ibmc.svn.sourceforge.net/viewvc/ibmc/IBMCDesk
- Installer: http://ibmc.svn.sourceforge.net/viewvc/ibmc/install/OpenTox_IBMC.rar
- Deployment requirements: Apache 2.63 or newer, PHP 5.0 or newer, MySQL 5.33 or newer
- Live demo: http://195.178.207.160/OpenTox/IBMCDesc


IBMC OpenTox descriptor calculation application - Mozilla Firefox
Файл Правка Вид Журнал Закладки Инструменты Справка
IBMC OpenTox descriptor calculation ap...
http://195.178.207.160/OpenTox/IBMCDesc
Яндекс
Самые популярные Начальная страница Лента новостей Status page QIP.RU: почта, поиск, ...
IBMC DESC.
MNA and QNA calculation
OpenTox descriptor calculation web app
Calculate
IBMC OpenTox

Please enter your dataset and choose calculated method:

Dataset URI*	
Dataset Service*	http://apps.ideaconsult.net:8080/ambit2/dataset/
Prediction feature*	
Select Algorithms	MNA
Select SCR model	MakeSCR_4

A&A*	
Login	
Password	
or	
Token	

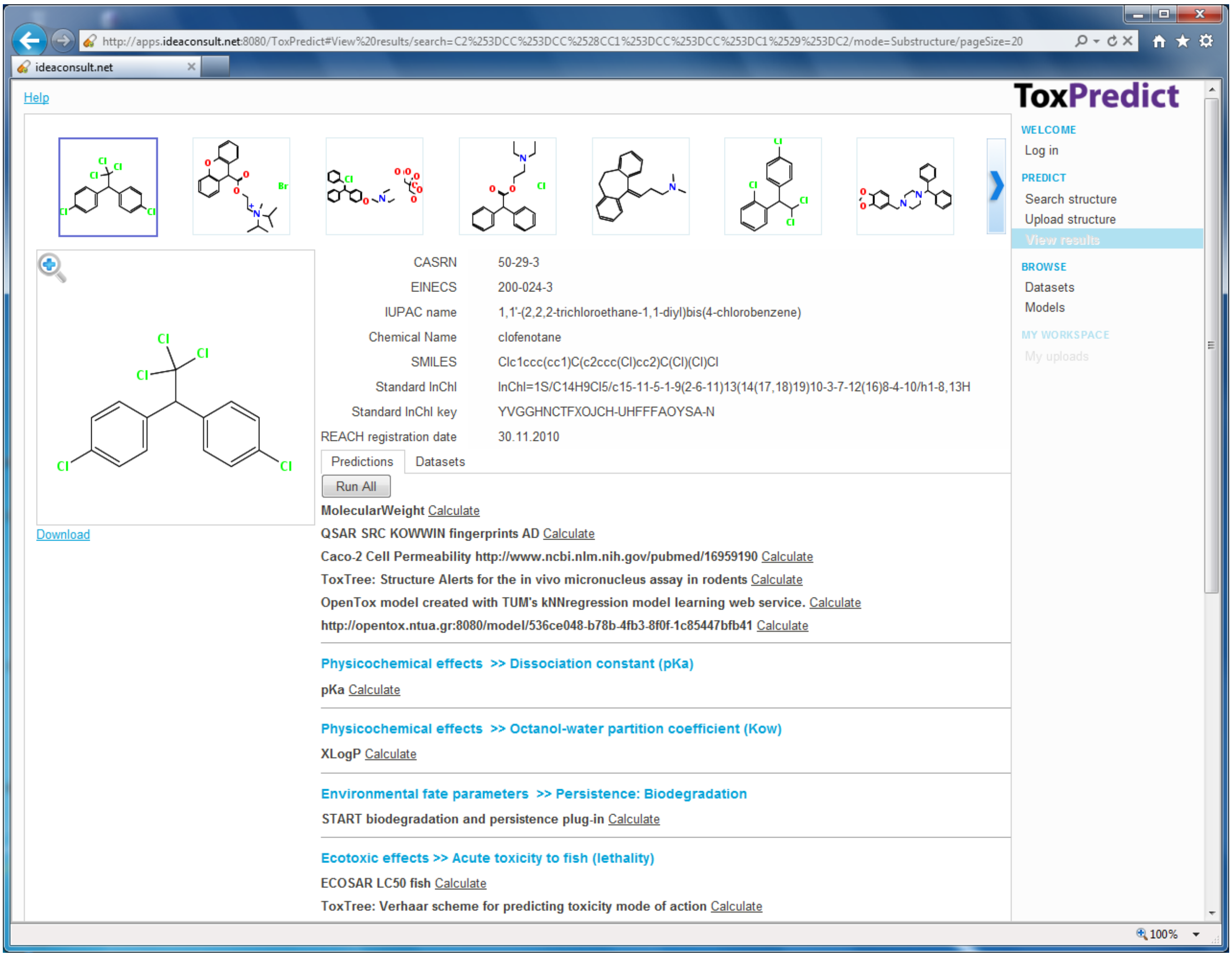
version 1.0
For questions please contact
dmitry.druzhilovsky@ibmc.msk.ru

Pharmaexpert.ru 2011

Figure 3 Screenshot of IBMCDesc application

4.10.4 ToxPredict application implemented by IDEA

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/svnroot/ambit/branches/opentox
- Binaries: http://ambit.sourceforge.net/downloads.html
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Live demo: http://toxpredict.org


http://apps.ideaconsult.net:8080/ToxPredict#View%20results/search=C2%253DCC%253DCC%2528CC1%253DCC%253DCC%253DC1%2529%253DC2/mode=Substructure/pageSize=20
ideaconsult.net
Help
ToxPredict
WELCOME
Log in
PREDICT
Search structure
Upload structure
View results
BROWSE
Datasets
Models
MY WORKSPACE
My uploads
Download
100%

CASRN	50-29-3
EINECS	200-024-3
IUPAC name	1,1'-(2,2,2-trichloroethane-1,1-diyl)bis(4-chlorobenzene)
Chemical Name	clofenotane
SMILES	Clc1ccc(cc1)C(c2ccc(Cl)cc2)C(Cl)(Cl)Cl
Standard InChI	InChI=1S/C14H9Cl5/c15-11-5-1-9(2-6-11)13(14(17,18)19)10-3-7-12(16)8-4-10/h1-8,13H
Standard InChI key	YVGGHNCTFXOJCH-UHFFFAOYSA-N
REACH registration date	30.11.2010

Predictions Datasets

Run All

MolecularWeight Calculate

QSAR SRC KOWWIN fingerprints AD Calculate

Caco-2 Cell Permeability http://www.ncbi.nlm.nih.gov/pubmed/16959190 Calculate

ToxTree: Structure Alerts for the in vivo micronucleus assay in rodents Calculate

OpenTox model created with TUM's kNNregression model learning web service. Calculate

http://opentox.ntua.gr:8080/model/536ce048-b78b-4fb3-8f0f-1c85447bfb41 Calculate

Physicochemical effects >> Dissociation constant (pKa)

pKa Calculate

Physicochemical effects >> Octanol-water partition coefficient (Kow)

XLogP Calculate

Environmental fate parameters >> Persistence: Biodegradation

START biodegradation and persistence plug-in Calculate

Ecotoxic effects >> Acute toxicity to fish (lethality)

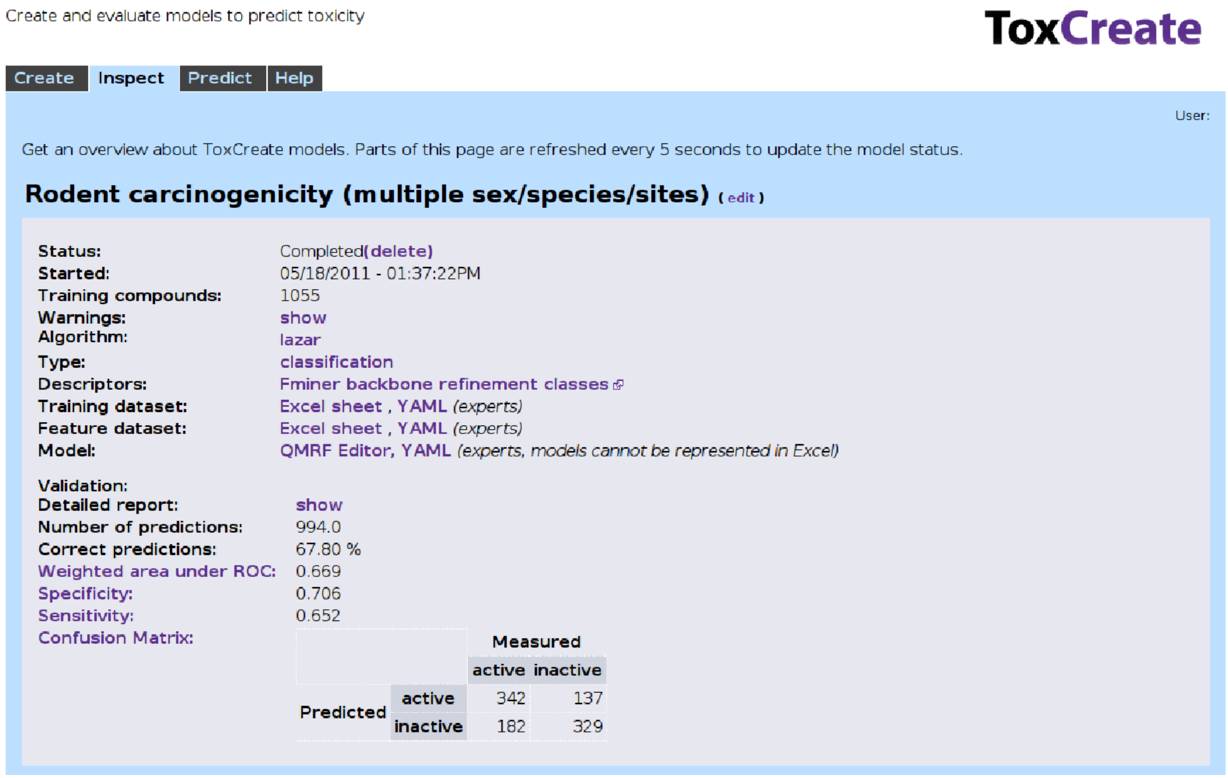
ECOSAR LC50 fish Calculate

ToxTree: Verhaar scheme for predicting toxicity mode of action Calculate

Figure 4 Screenshot of the ToxPredict web application

4.10.5 ToxCreate application implemented by IST

- Programming language: Ruby
- Source code repository: http://github.com/opentox/toxcreate/
- Installer: http://github.com/opentox/install/
- Deployment requirements: Debian-based Linux distribution
- Live demo: http://toxcreate.org



Get an overview about ToxCreate models. Parts of this page are refreshed every 5 seconds to update the model status.

Rodent carcinogenicity (multiple sex/species/sites) (edit)

Status:	Completed(delete)
Started:	05/18/2011 - 01:37:22PM
Training compounds:	1055
Warnings:	show
Algorithm:	lazar
Type:	classification
Descriptors:	Fminer backbone refinement classes
Training dataset:	Excel sheet , YAML *(experts)*
Feature dataset:	Excel sheet , YAML *(experts)*
Model:	QMRF Editor, YAML *(experts, models cannot be represented in Excel)*

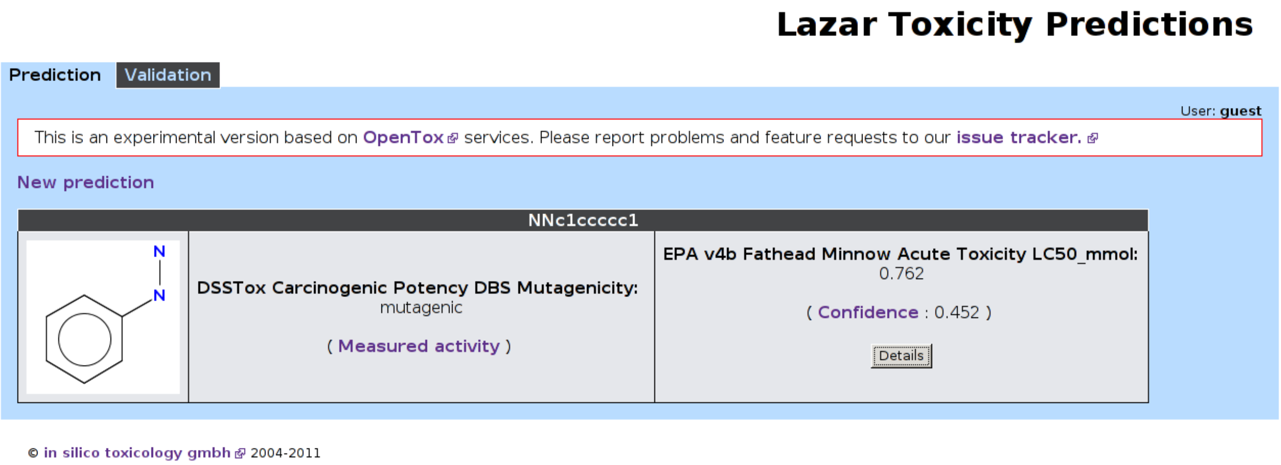
Validation:	
Detailed report:	show
Number of predictions:	994.0
Correct predictions:	67.80 %
Weighted area under ROC:	0.669
Specificity:	0.706
Sensitivity:	0.652
Confusion Matrix:	

		Measured	
		active	inactive
Predicted	active	342	137
	inactive	182	329

Figure 5 Screenshot of the ToxCreate web application

4.10.6 lazar application implemented by IST (CH)

- Programming language: Ruby
- Source code repository: http://github.com/opentox/toxcreate/ (branch lazar-gui)
- Installer: http://github.com/opentox/install/
- Deployment requirements: Debian-based Linux distribution
- Live demo: http://lazar.in-silico.ch/



Lazar Toxicity Predictions

User: **guest**

This is an experimental version based on **OpenTox** services. Please report problems and feature requests to our **issue tracker.**

New prediction

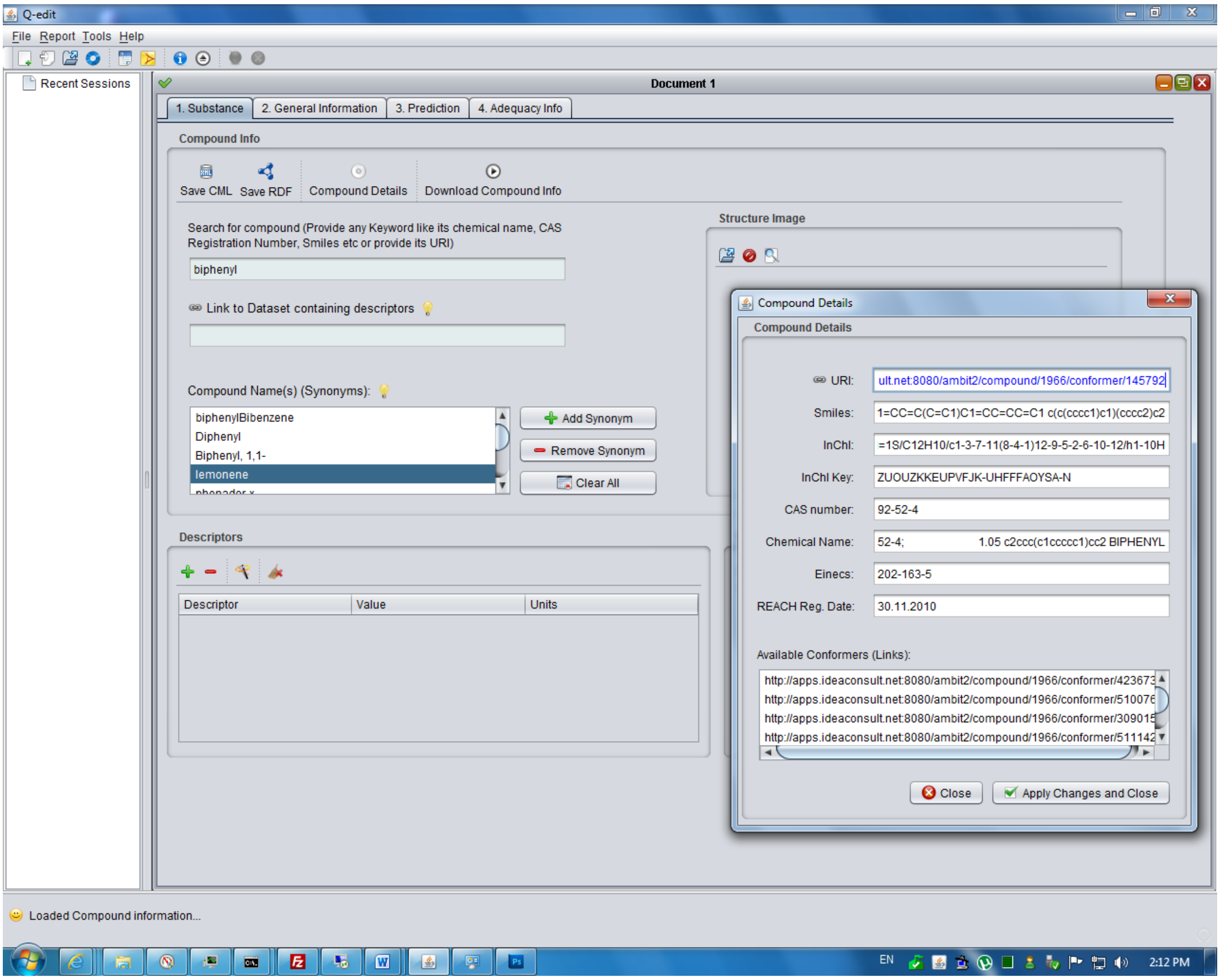
NNc1ccccc1		
	DSSTox Carcinogenic Potency DBS Mutagenicity: mutagenic (**Measured activity**)	**EPA v4b Fathead Minnow Acute Toxicity LC50_mmol:** 0.762 (**Confidence** : 0.452) Details

© **in silico toxicology gmbh** 2004-2011

Figure 6: Screenshot of lazar predictions

4.10.7 Q-Edit application implemented by NTUA

- Programming language: Java
- Source code repository: https://github.com/alphaville/Q-edit
- Installer: https://github.com/alphaville/Q-edit/downloads
- Dependencies and requirements: JVM version 1.6 or higher (all other dependencies are shipped with the binaries)



Search for compound (Provide any Keyword like its chemical name, CAS Registration Number, Smiles etc or provide its URI): biphenyl

Link to Dataset containing descriptors:

Compound Name(s) (Synonyms):
biphenylBibenzene
Diphenyl
Biphenyl, 1,1-
lemonene
phenador-x

Descriptor	Value	Units

URI: ult.net:8080/ambit2/compound/1966/conformer/145792
Smiles: 1=CC=C(C=C1)C1=CC=CC=C1 c(c(cccc1)c1)(cccc2)c2
InChI: =1S/C12H10/c1-3-7-11(8-4-1)12-9-5-2-6-10-12/h1-10H
InChI Key: ZUOUZKKEUPVFJK-UHFFFAOYSA-N
CAS number: 92-52-4
Chemical Name: 52-4; 1.05 c2ccc(c1ccccc1)cc2 BIPHENYL
Einecs: 202-163-5
REACH Reg. Date: 30.11.2010

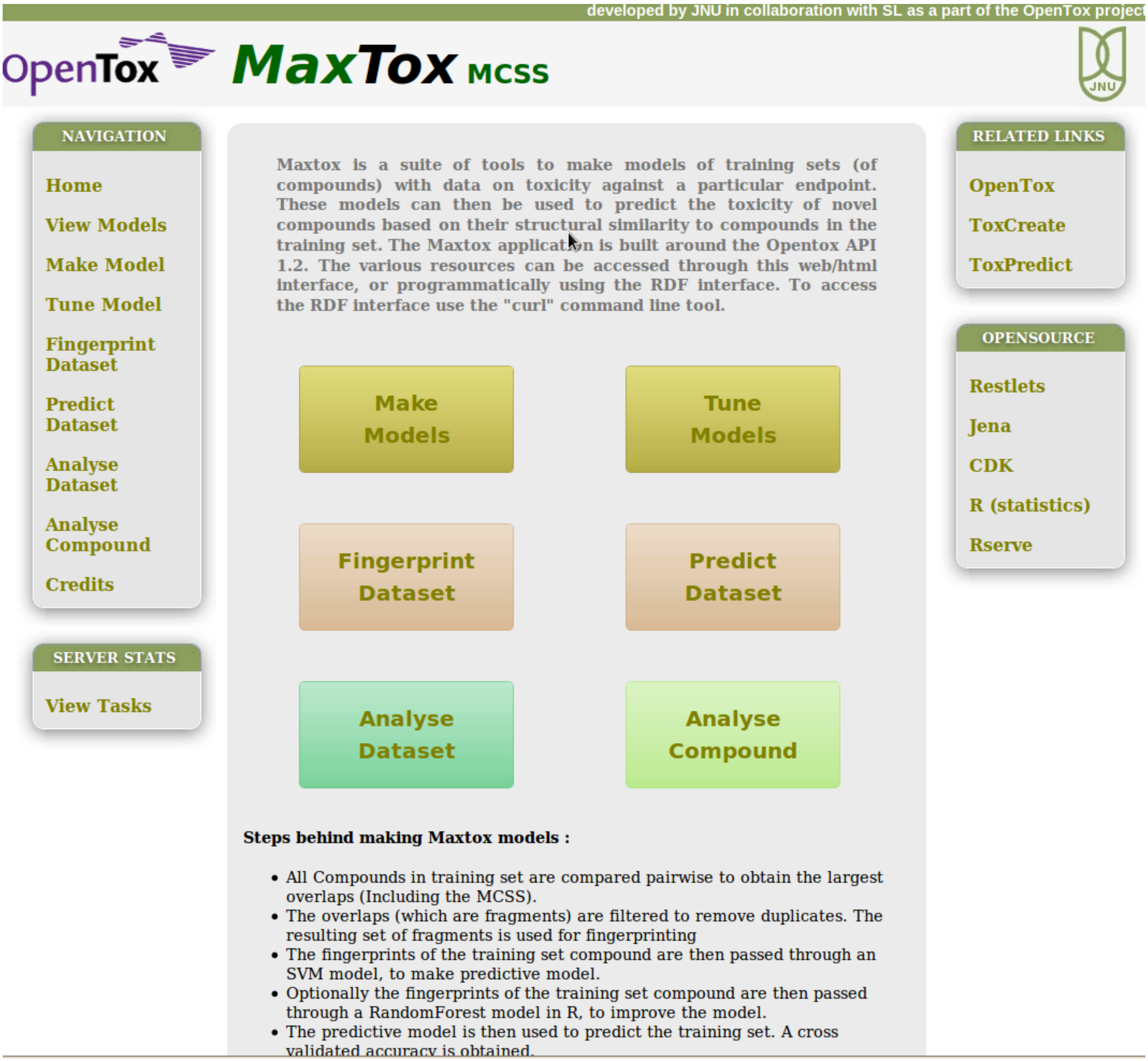
Available Conformers (Links):
http://apps.ideaconsult.net:8080/ambit2/compound/1966/conformer/423673
http://apps.ideaconsult.net:8080/ambit2/compound/1966/conformer/510076
http://apps.ideaconsult.net:8080/ambit2/compound/1966/conformer/309015
http://apps.ideaconsult.net:8080/ambit2/compound/1966/conformer/511142

Loaded Compound information...

Figure 7 Screenshot of the Q-Edit Java application

4.10.8 MaxToxMCSS application implemented by SL-JNU

- Programming language: Java
- Source code: http://opentox2.informatik.uni-freiburg.de/files/MaxtoxMCSS.src.zip
- Installer: http://opentox2.informatik.uni-freiburg.de/files/MaxtoxMCSS.war
- Dependencies : cdk-1.3.4, restlet-2.1, jena-2.6.2, jchempaint-8, JNI_SVM-light-6.01, Rserve
- Deployment requirements: Apache Tomcat 6.0 or newer, MySQL 5.1 or newer
- Live demo: http://opentox2.informatik.uni-freiburg.de:8080/MaxtoxMCSS/

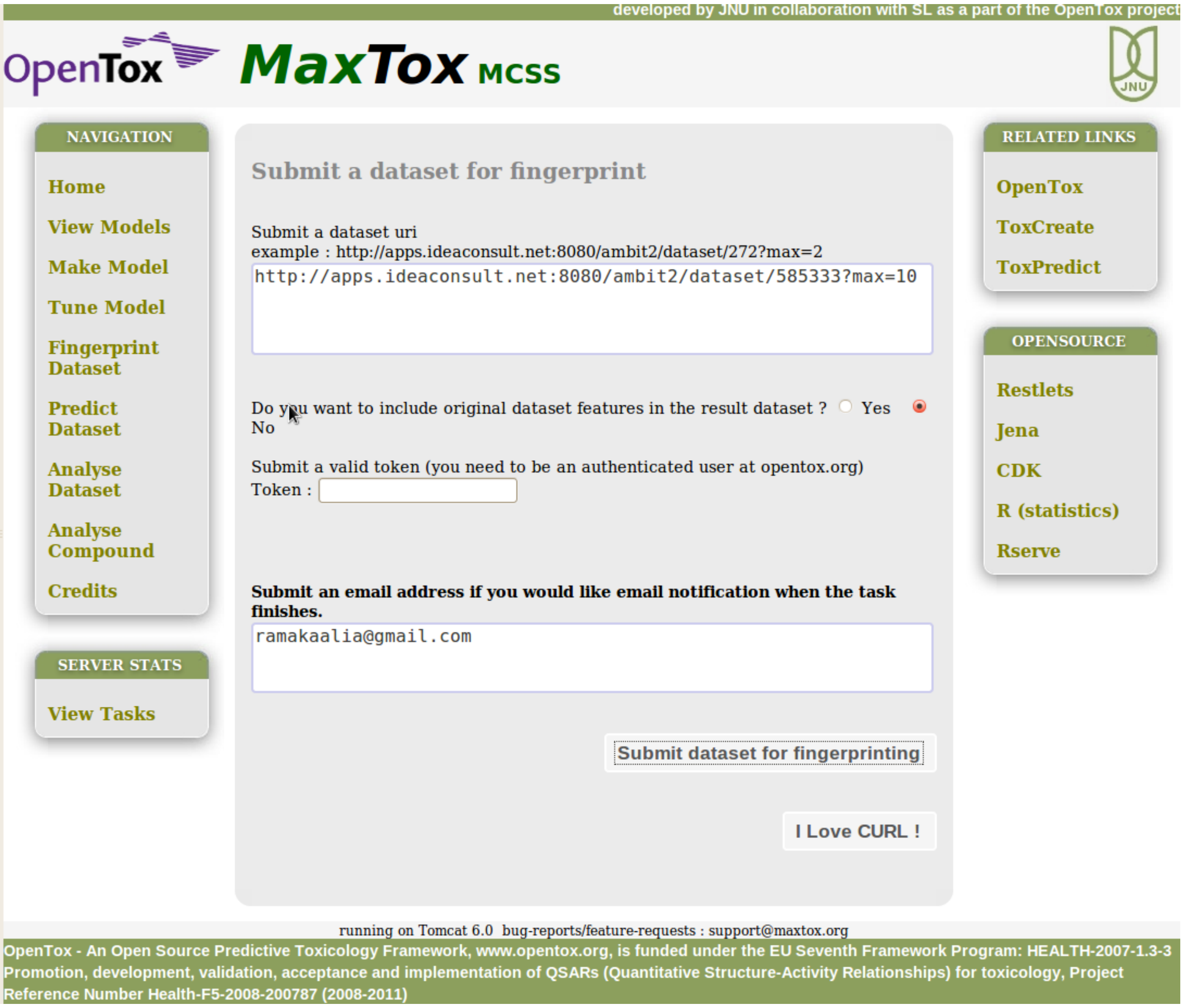


Maxtox is a suite of tools to make models of training sets (of compounds) with data on toxicity against a particular endpoint. These models can then be used to predict the toxicity of novel compounds based on their structural similarity to compounds in the training set. The Maxtox application is built around the Opentox API 1.2. The various resources can be accessed through this web/html interface, or programmatically using the RDF interface. To access the RDF interface use the "curl" command line tool.

Steps behind making Maxtox models :

- All Compounds in training set are compared pairwise to obtain the largest overlaps (Including the MCSS).
- The overlaps (which are fragments) are filtered to remove duplicates. The resulting set of fragments is used for fingerprinting
- The fingerprints of the training set compound are then passed through an SVM model, to make predictive model.
- Optionally the fingerprints of the training set compound are then passed through a RandomForest model in R, to improve the model.
- The predictive model is then used to predict the training set. A cross validated accuracy is obtained.

Figure 8 Screenshot of MaxToxMCSS web application – home page



Submit a dataset for fingerprint

Submit a dataset uri
example : http://apps.ideaconsult.net:8080/ambit2/dataset/272?max=2

```
http://apps.ideaconsult.net:8080/ambit2/dataset/585333?max=10
```

Do you want to include original dataset features in the result dataset ? ○ Yes ● No

Submit a valid token (you need to be an authenticated user at opentox.org)
Token :

Submit an email address if you would like email notification when the task finishes.

```
ramakaalia@gmail.com
```

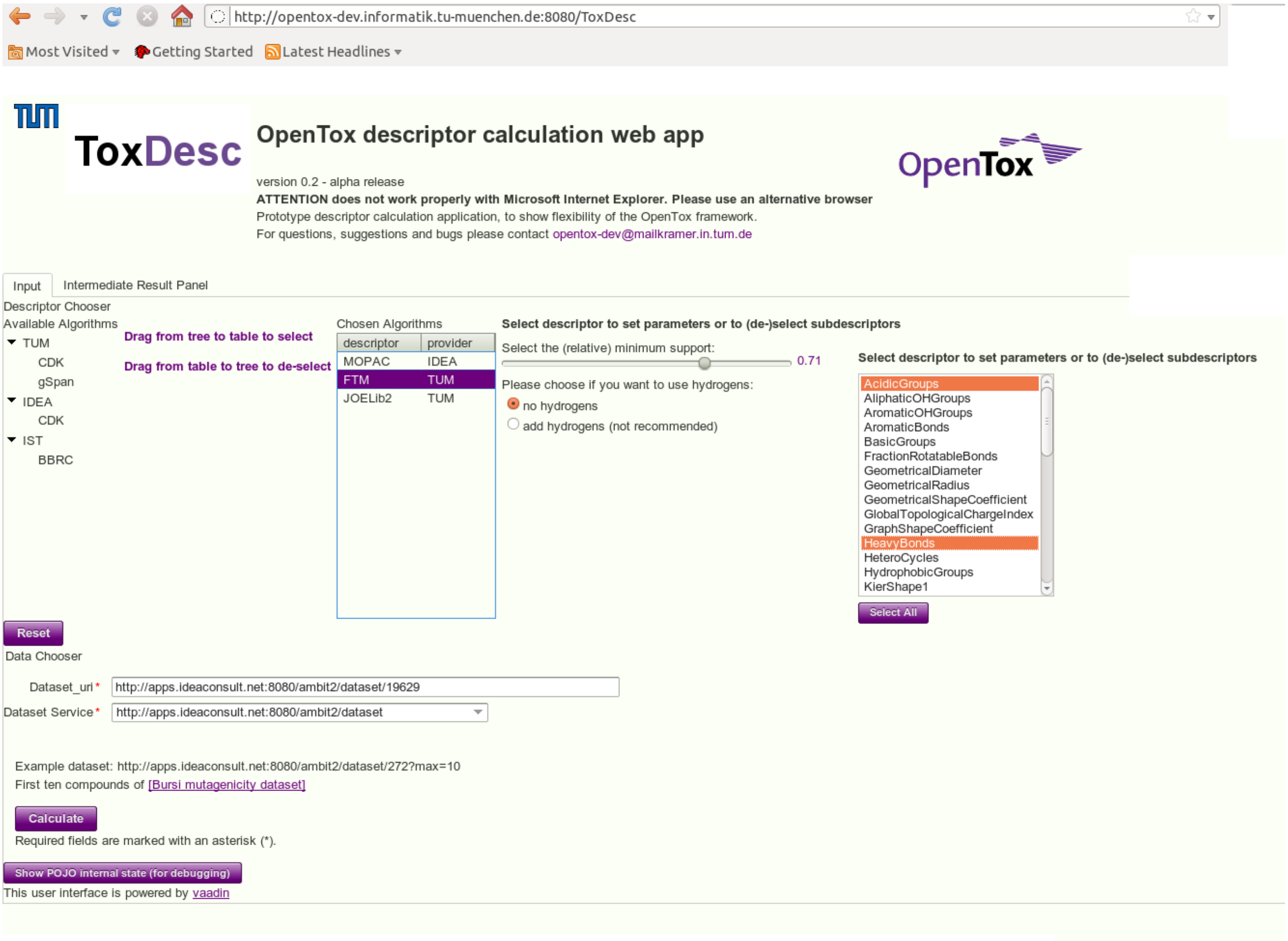
running on Tomcat 6.0 bug-reports/feature-requests : support@maxtox.org

OpenTox - An Open Source Predictive Toxicology Framework, www.opentox.org, is funded under the EU Seventh Framework Program: HEALTH-2007-1.3-3 Promotion, development, validation, acceptance and implementation of QSARs (Quantitative Structure-Activity Relationships) for toxicology, Project Reference Number Health-F5-2008-200787 (2008-2011)

Figure 9 Screenshot of MaxToxMCSS web application – fingerprints generation

4.10.9 ToxDesc application implemented by TUM

- Programming language: Java
- Source code repository: http://opentox.informatik.tu-muenchen.de/trac/TUMOpenTox/browser/DescCalcGUI/src
- Binaries: http://opentox.informatik.tu-muenchen.de/trac/TUMOpenTox/wiki/Download
- Deployment requirements: Apache Tomcat 6.0 or newer
- Live demo: http://opentox-dev.informatik.tu-muenchen.de:8080/ToxDesc


http://opentox-dev.informatik.tu-muenchen.de:8080/ToxDesc
Most Visited
Getting Started
Latest Headlines
Input
Intermediate Result Panel
Reset
Select All
Calculate
Show POJO internal state (for debugging)

ToxDesc

OpenTox descriptor calculation web app

version 0.2 - alpha release

ATTENTION does not work properly with Microsoft Internet Explorer. Please use an alternative browser

Prototype descriptor calculation application, to show flexibility of the OpenTox framework.

For questions, suggestions and bugs please contact opentox-dev@mailkramer.in.tum.de

Descriptor Chooser

Available Algorithms

- TUM
  - CDK
  - gSpan
- IDEA
  - CDK
- IST
  - BBRC

Drag from tree to table to select

Drag from table to tree to de-select

Chosen Algorithms

descriptor	provider
MOPAC	IDEA
FTM	TUM
JOELib2	TUM

Select descriptor to set parameters or to (de-)select subdescriptors

Select the (relative) minimum support: 0.71

Please choose if you want to use hydrogens:

● no hydrogens

○ add hydrogens (not recommended)

Select descriptor to set parameters or to (de-)select subdescriptors

AcidicGroups
AliphaticOHGroups
AromaticOHGroups
AromaticBonds
BasicGroups
FractionRotatableBonds
GeometricalDiameter
GeometricalRadius
GeometricalShapeCoefficient
GlobalTopologicalChargeIndex
GraphShapeCoefficient
HeavyBonds
HeteroCycles
HydrophobicGroups
KierShape1

Data Chooser

Dataset_uri* http://apps.ideaconsult.net:8080/ambit2/dataset/19629

Dataset Service* http://apps.ideaconsult.net:8080/ambit2/dataset

Example dataset: http://apps.ideaconsult.net:8080/ambit2/dataset/272?max=10
First ten compounds of [Bursi mutagenicity dataset]

Required fields are marked with an asterisk (*).

This user interface is powered by vaadin

Figure 10 Screenshot of the ToxDesc web application

4.10.10 Taverna workflow implemented by TUM

- Requirements: Taverna (available at http://www.taverna.org.uk/)
- Workflow example available at: http://www.myexperiment.org/tags/1738.html

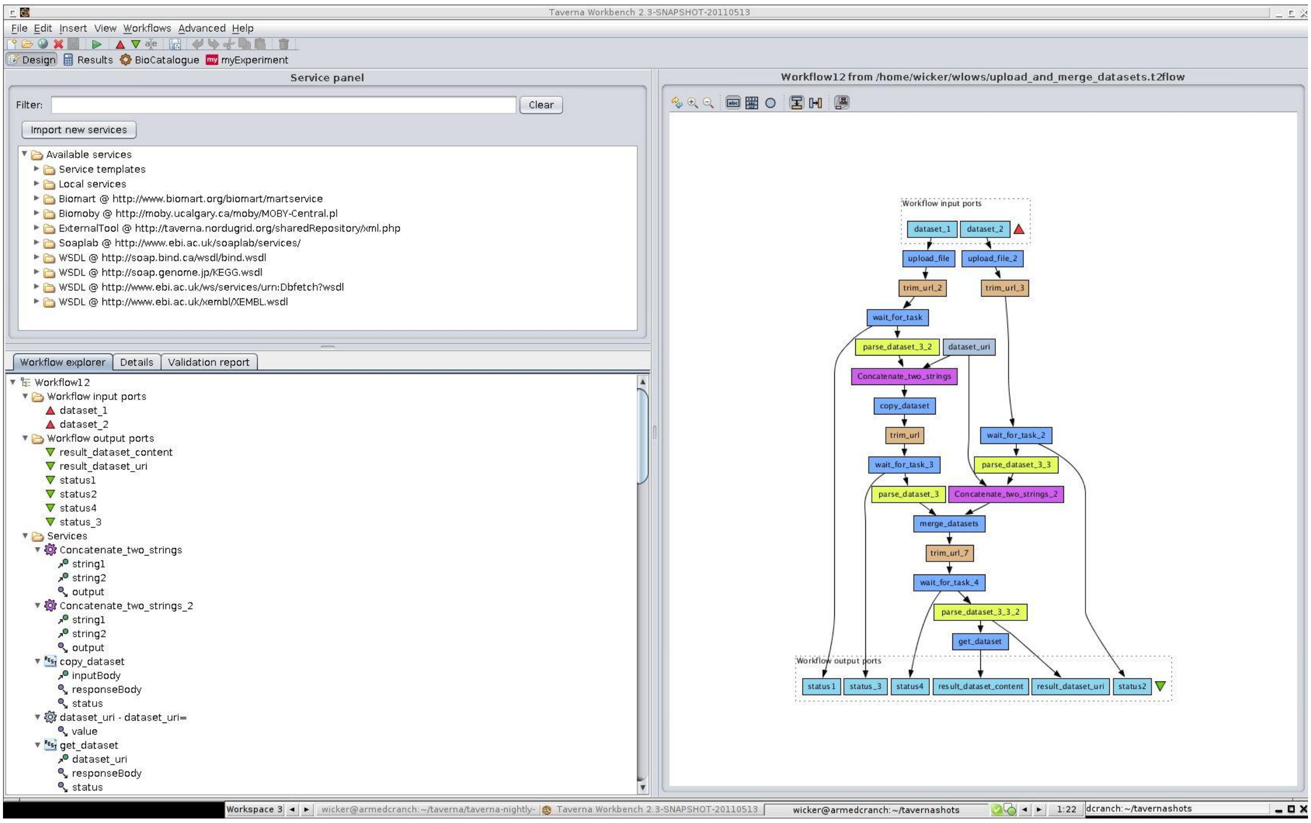

Taverna Workbench 2.3-SNAPSHOT-20110513
File Edit Insert View Workflows Advanced Help
Design Results BioCatalogue myExperiment
Service panel
Filter:
Clear
Import new services
Available services
Service templates
Local services
Biomart @ http://www.biomart.org/biomart/martservice
Biomoby @ http://moby.ucalgary.ca/moby/MOBY-Central.pl
ExternalTool @ http://taverna.nordugrid.org/sharedRepository/xml.php
Soaplab @ http://www.ebi.ac.uk/soaplab/services/
WSDL @ http://soap.bind.ca/wsdl/bind.wsdl
WSDL @ http://soap.genome.jp/KEGG.wsdl
WSDL @ http://www.ebi.ac.uk/ws/services/urn:Dbfetch?wsdl
WSDL @ http://www.ebi.ac.uk/xembl/XEMBL.wsdl
Workflow explorer Details Validation report
Workflow12
Workflow input ports
dataset_1
dataset_2
Workflow output ports
result_dataset_content
result_dataset_uri
status1
status2
status4
status_3
Services
Concatenate_two_strings
string1
string2
output
Concatenate_two_strings_2
string1
string2
output
copy_dataset
inputBody
responseBody
status
dataset_uri - dataset_uri=
value
get_dataset
dataset_uri
responseBody
status
Workflow12 from /home/wicker/wlows/upload_and_merge_datasets.t2flow
Workflow input ports
dataset_1
dataset_2
upload_file
upload_file_2
trim_url_2
trim_url_3
wait_for_task
parse_dataset_3_2
dataset_uri
Concatenate_two_strings
copy_dataset
trim_url
wait_for_task_2
wait_for_task_3
parse_dataset_3_3
parse_dataset_3
Concatenate_two_strings_2
merge_datasets
trim_url_7
wait_for_task_4
parse_dataset_3_3_2
get_dataset
Workflow output ports
status1
status_3
status4
result_dataset_content
result_dataset_uri
status2
Workspace 3
wicker@armedcranch:~/taverna/taverna-nightly-
Taverna Workbench 2.3-SNAPSHOT-20110513
wicker@armedcranch:~/tavernashots
1:22
dcranch:~/tavernashots

Figure 11 Screenshot of Taverna workflow

4.10.11 Bioclipse application with OpenTox API implemented by 3rd party

Bioclipse[58,59,60] is a free and open source workbench for the life sciences. It provides advanced functionality in fields such as cheminformatics, bioinformatics, semantic web, spectrum analysis, drug discovery, safety assessment, and general chemistry education. Bioclipse is developed as a collaboration between the Proteochemometric Group, Dept. of Pharmaceutical Biosciences, Uppsala University, Sweden[61], and the Cheminformatics and Metabolism Team at the European Bioinformatics Institute (EBI)[62]. Bioclipse-OpenTox is a set of plugins which adds functionality to Bioclipse for interacting with the OpenTox infrastructure. Features include uploading to and downloading data from OpenTox servers, discovery of new computational toxicology resources on the OpenTox network, and the combined performance of local and remote computations of predictions. Binaries are available for:

- Windows (32-bit): http://pele.farmbio.uu.se/dl/bioclipse2.5-DS-OT-QSAR-win32.win32.x86.zip
- Windows (64-bit): http://pele.farmbio.uu.se/dl/bioclipse2.5-DS-OT-QSAR-win32.win32.x86_64.zip
- Mac OS X (64-bit): http://pele.farmbio.uu.se/dl/bioclipse2.5-DS-OT-QSAR-macosx.cocoa.x86_64.zip
- Linux (32-bit): http://pele.farmbio.uu.se/dl/bioclipse2.5-DS-OT-QSAR-linux.gtk.x86.zip
- Linux (64-bit): http://pele.farmbio.uu.se/dl/bioclipse2.5-DS-OT-QSAR-linux.gtk.x86_64.zip

The source code of the Bioclipse-OpenTox set of plugins is available at https://github.com/bioclipse/bioclipse.opentox

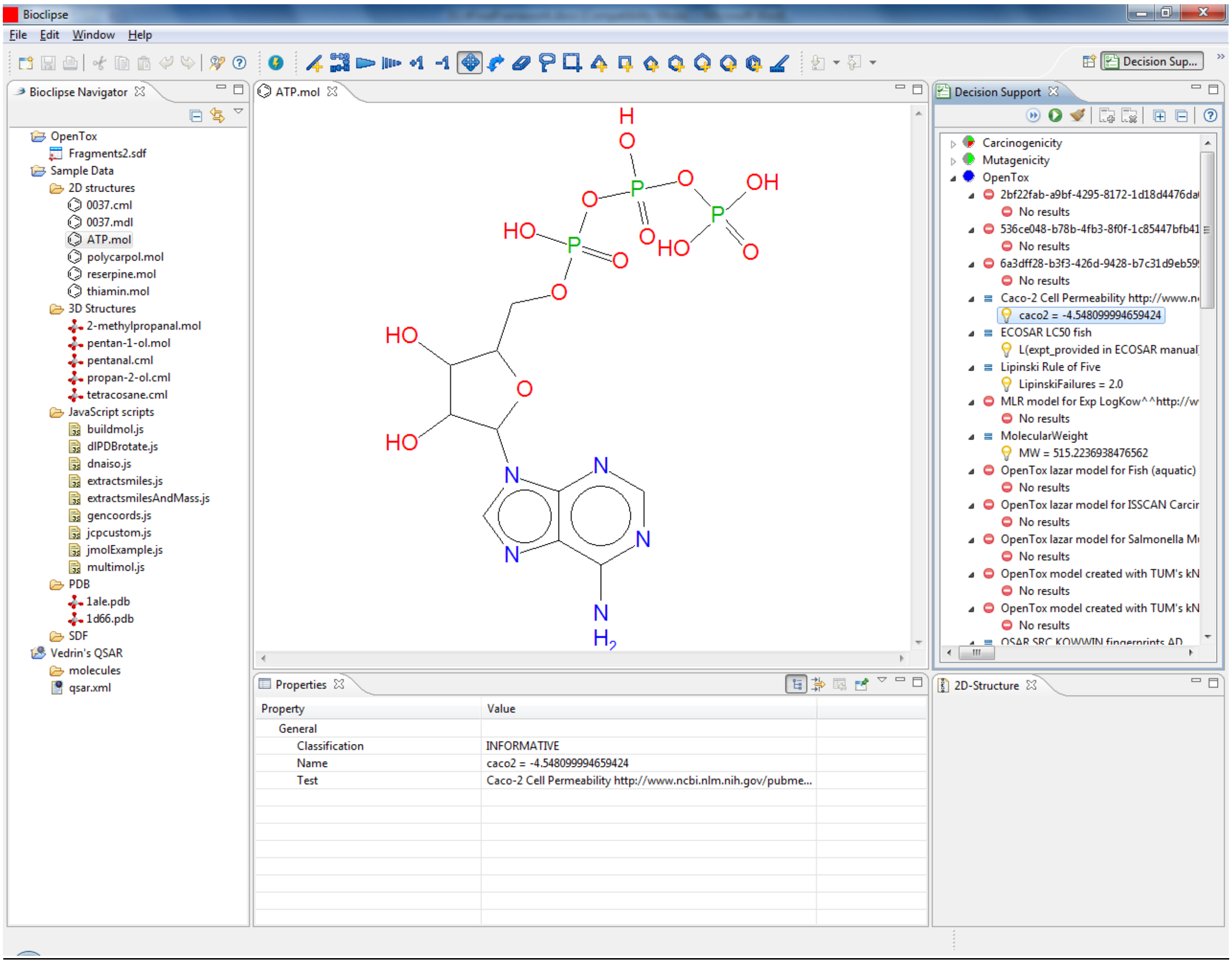
[58] Bioclipse: An open source workbench for chemo- and bioinformatics. Ola Spjuth, Tobias Helmus, Egon L Willighagen, Stefan Kuhn, Martin Eklund, Johannes Wagener, Peter Murray-Rust, Christoph Steinbeck, Jarl E.S. Wikberg BMC Bioinformatics 2007, 8:59 doi:10.1186/1471-2105-8-59

[59] Bioclipse 2: A scriptable integration platform for the life sciences. Ola Spjuth, Jonathan Alvarsson, Arvid Berg, Martin Eklund, Stefan Kuhn, Carl MÃ¤sak, Gilleain Torrance, Johannes Wagener, Egon L Willighagen, Christoph Steinbeck and Jarl ES Wikberg BMC Bioinformatics 2009, 10:397 doi:10.1186/1471-2105-10-397

[60] http://www.bioclipse.net/ accessed on June 7, 2011

[61] http://www.farmbio.uu.se/researchgroup.php?fg=1 accessed on June 7, 2011

[62] http://www.ebi.ac.uk/steinbeck/ accessed on June 7, 2011



Carcinogenicity
Mutagenicity
OpenTox
2bf22fab-a9bf-4295-8172-1d18d4476da
No results
536ce048-b78b-4fb3-8f0f-1c85447bfb41
No results
6a3dff28-b3f3-426d-9428-b7c31d9eb59
No results
Caco-2 Cell Permeability http://www.n
caco2 = -4.548099994659424
ECOSAR LC50 fish
L(expt_provided in ECOSAR manual
Lipinski Rule of Five
LipinskiFailures = 2.0
MLR model for Exp LogKow^^http://w
No results
MolecularWeight
MW = 515.2236938476562
OpenTox lazar model for Fish (aquatic)
No results
OpenTox lazar model for ISSCAN Carcir
No results
OpenTox lazar model for Salmonella M
No results
OpenTox model created with TUM's kN
No results
OpenTox model created with TUM's kN
No results
QSAR SRC KOWWIN fingerprints AD

Properties

Property	Value
General	
Classification	INFORMATIVE
Name	caco2 = -4.548099994659424
Test	Caco-2 Cell Permeability http://www.ncbi.nlm.nih.gov/pubme...

Figure 12 Screenshot of Bioclipse displaying results from OpenTox models

A recording of a Bioclipse-OpenTox tutorial session which took place on May 25, 2011, is available at http://www.opentox.org/data/documents/development/tutorialfiles/downloads/2011-05-25_Bioclipse-OpenTox-Integration.wmv/view. Some further instructions related to this seminar are available at http://wiki.bioclipse.net/index.php?title=OpenTox_Online_Seminar

4.10.12 Bioclipse workflows implemented by 3rd party

A set of Bioclipse scripting language scripts for accessing and working with the OpenTox framework, developed by Egon Willighagen, are available at http://www.myexperiment.org/search?query=opentox+bioclipse&type=workflows. These scripts perform the following tasks, when executed in the Bioclipse rich client application:

- List all algorithms and descriptors an OpenTox service provides (http://www.myexperiment.org/workflows/1204.html);
- Download QSAR data sets using the OpenTox API 1.1 (http://www.myexperiment.org/workflows/1192.html);
- Calculate molecular descriptors with OpenTox services (http://www.myexperiment.org/workflows/1646.html);

- Download all data sets from an OpenTox API capable web resource (http://www.myexperiment.org/workflows/1008.html);
- Create an OpenTox API dataset and add molecules (http://www.myexperiment.org/workflows/1468.html);
- Run OpenTox models on local data (http://www.myexperiment.org/workflows/1658.html);
- Search OpenTox data sets by title (http://www.myexperiment.org/workflows/1660.html);
- Authenticate with the OpenTox network (http://www.myexperiment.org/workflows/2149.html).

4.11 OpenTox client libraries

OpenTox libraries are provided to third-party software developers in order to ease the integration of their tools with relevant OpenTox web services. This section presents four client libraries, developed by IDEA, IST and NTUA.

4.11.1 OT-client library implemented by IDEA

The OT-client library implements Java classes that encapsulate client access to the OpenTox API. It relies on the OT-sso library (described below) for AA-related queries.

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/svnroot/ambit/branches/opentox/opentox-dsl
- Binaries (Maven repository): http://ambit.uni-plovdiv.bg:8083/nexus/index.html#nexus-search;quick~opentox-dsl
- Platform requirements: Java 6

4.11.2 OT-sso client library implemented by IDEA

The OT-sso client libraries implements Java classes that encapsulate client access to the OpenTox authentication, authorisation and policy management API.

- Programming language: Java
- Source code repository: http://ambit.svn.sourceforge.net/svnroot/ambit/trunk/opentox-opensso
- Binaries (Maven repository): http://ambit.uni-plovdiv.bg:8083/nexus/index.html#nexus-search;quick~opentox-opensso
- Platform requirements: Java 6

4.11.3 OpenTox-ruby client library implemented by IST

- Programming language: Ruby
- Source code repository: https://github.com/opentox/opentox-ruby
- Usage: opentox-ruby depends on many third-party programs and libraries, which makes the setup complicated and error prone. For this reason we recommend to use the installer from https://github.com/opentox/install. If you want to install manually you can find the necessary steps in the installation scripts.

4.11.4 ToxOtis client library implemented by NTUA

- Programming language: Java

- Source code repository: https://github.com/alphaville/ToxOtis
- Maven repository (includes binaries, Javadoc and source code artifacts): http://ambit.uni-plovdiv.bg:8083/nexus/content/repositories/thirdparty/org/opentox/toxotis/
- About: ToxOtis is the Greek word for Sagittarius, that actually means 'archer'. ToxOtis is a Java interface to the predictive toxicology services of OpenTox. ToxOtis is being developed to help both those who need a painless way to consume OpenTox web services and for ambitious service providers that don't want to spend half of their time in RDF parsing and creation, database management and security measures. You can use ToxOtis to search in databases with chemical compounds, download a compound in any supported MIME type (e.g. SDF, SMILES, MOL etc.), find a property for a compound (e.g. its LD50[63] lethal dose concentration), publish your chemicals in an online database, train QSAR models (regression, classification, clustering etc.) and lots of other functionalities. Incorporation of ToxOtis into your services will relieve you from the labour of creating RDF documents.
- Documentation: https://github.com/alphaville/ToxOtis/tree/master/doc

4.12 OpenTox virtual appliance

The IST OpenTox Virtual Appliance (VA), built under VirtualBox[64] by IST, was extended to include the services developed by IDEA as well as the final OpenTox database (see deliverable report "D3.4 Final database with additional content"). First, the IST OpenTox VA was downloaded and set up as described at http://www.maunz.de/wordpress/opentox/2011/using-ist-opentox-virtual-appliance-under-windows. Additional packages were installed to the VA (most importantly MySQL and Tomcat) using the package manager of the virtual appliance. The Ambit REST services, the ontology service and ToxPredict were downloaded from http://ambit.uni-plovdiv.bg/downloads/ambit2/ and installed following the accompanying instructions. The downloadable archives for Ambit and ToxPredict have been configured to expect the ontology service at http://localhost:8080/ontology, resulting in a VA in which these services function without requiring an Internet connection.

The database was also downloaded from http://ambit.uni-plovdiv.bg/downloads/ambit2/ as an SQL dump. The downloaded file was stored on a newly created additional hard disk within the VA, and imported in MySQL. After having successfully imported the database, the additional hard disk was again deleted from the appliance. A set of 16 models were created inside the local installation of the AMBIT software and registered with the local installation of the ontology service.

Q-edit was downloaded from https://github.com/alphaville/Q-edit/downloads/, installed on the VA, and a shortcut for launching the application was placed on the desktop of the virtual appliance.

As a result, a significant subset of core OpenTox functionalities, developed by different partners by using different programming languages and development tools/libraries, have been packaged together in the virtual appliance and made available in a very convenient way for demo and experimental purposes. These tools are compatible with each other and rely on the OpenTox APIs and ontologies to orchestrate their work. The appliance can be downloaded from the following locations (mirror sites):

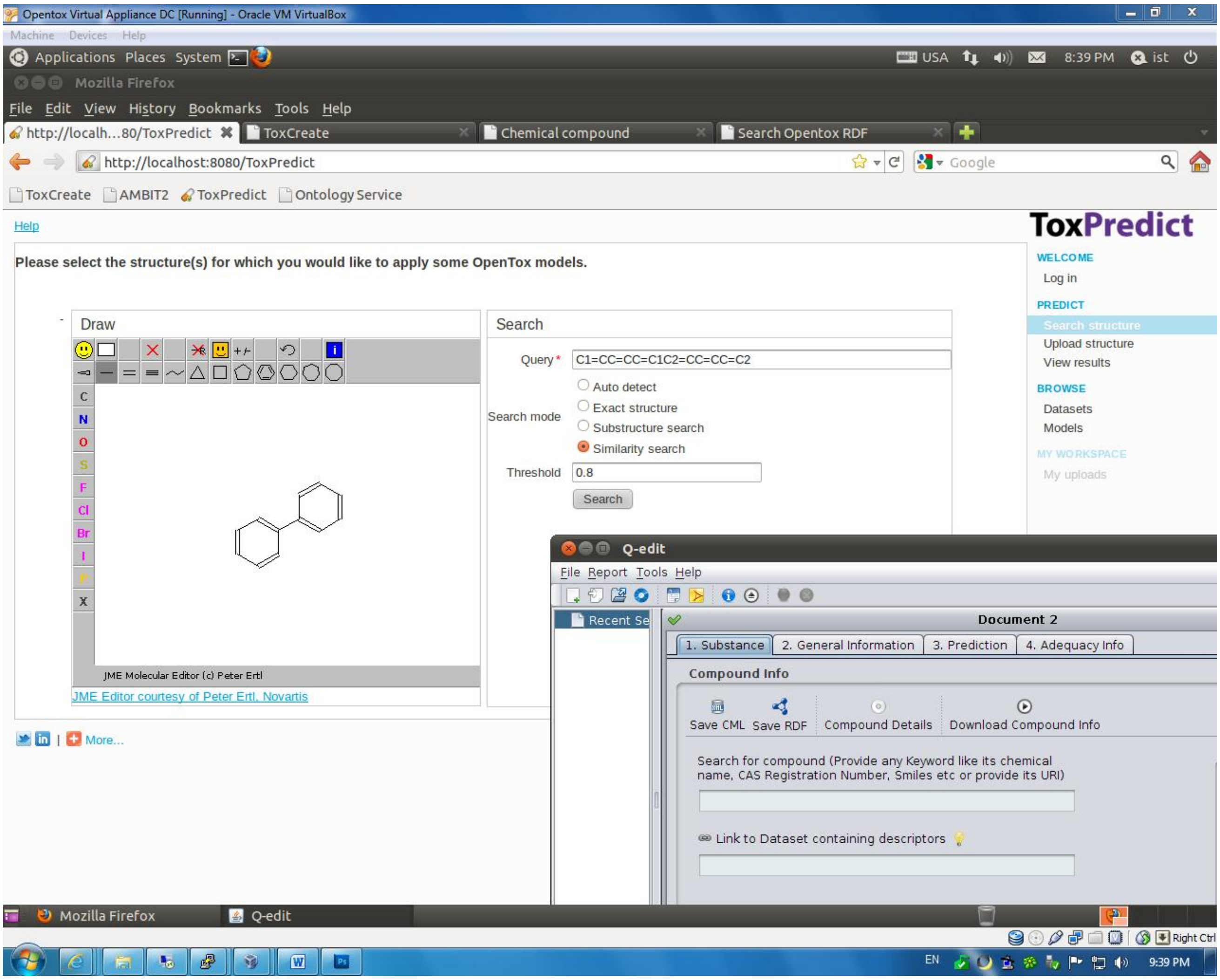
- http://ambit.uni-plovdiv.bg/downloads/opentox/Opentox Virtual Appliance DC.ova
- http://www.ideaconsult.net/downloads/opentox/Opentox Virtual Appliance DC.ova

[63] http://en.wikipedia.org/wiki/Median_lethal_dose accessed on June 9, 2011

[64] http://www.virtualbox.org/ accessed on May 31, 2011

Please note that this is a large file (2730474496 bytes). Its md5 checksum which you could check to ensure that no errors have occurred while downloading it is: 1530bb83e88c3c646bcbac3183745bab

Figure 13 illustrates the virtual appliance running in VirtualBox on a 64-bit Windows 7 host operating system and accessing the built-in local instances of ToxPredict, ToxCreate, Q-Edit, as well as ambit2 and ontology web services.



Opentox Virtual Appliance DC [Running] - Oracle VM VirtualBox

http://localhost:8080/ToxPredict

Please select the structure(s) for which you would like to apply some OpenTox models.

Query*: C1=CC=CC=C1C2=CC=CC=C2

Search mode: ○ Auto detect ○ Exact structure ○ Substructure search ● Similarity search

Threshold: 0.8

JME Molecular Editor (c) Peter Ertl

JME Editor courtesy of Peter Ertl, Novartis

Q-edit — Document 2

Compound Info

Search for compound (Provide any Keyword like its chemical name, CAS Registration Number, Smiles etc or provide its URI)

Link to Dataset containing descriptors

Figure 13 OpenTox virtual appliance

5. Conclusions

The services – implementing the OpenTox API for compounds, datasets, and features – enable importing arbitrary files with chemical structures and their properties, allowing linking to computer readable information about the data fields as well as keeping provenance information. In addition, they support multiple structures of the same compound, which is useful for storing and working with multiple conformations, as well as for comparing structures originally residing in different source databases. Uploading a file with chemical structures and data makes it automatically available in several formats, including the mandatory RDF representation, defined by the OpenTox ontology. The services, implementing the OpenTox API for algorithms and models, provide a unified web service interface to several descriptor calculation, machine learning, and structure searching algorithms, as well as to applicability domain and toxicity prediction models. The complexity and diversity of the processing is reduced to the simple paradigm "read data from a web address, perform processing, write to a web address”. The online services allow running predictions without installing any software, as well as sharing datasets and models between online users. The downloadable applications allow researchers to set up their own systems of chemical compounds, calculated and experimental data, and to run existing algorithms and create new models. The advantage of exposing the functionality via the OpenTox API is that all these resources could interoperate seamlessly, not only within a single web application, but also in a network of many cooperating distributed services.

Exposing functionalities through a web application programming interface allows the developer to hide the implementation details of both data storage (different database types vs. memory vs. file system backend) and processing (descriptor calculation algorithms using The Chemistry Development Kit[65,66,67], OpenBabel[68], commercial or in-house implementations). The availability of data and processing resources as RDF facilitates integrates the resources as linked data. The distributed algorithm and model resources automatically generate RDF representations, making the linked data dynamic, and not relying on a single traditional triple storage. The classes in the OpenTox ontology are designed to cover the minimum number of building blocks necessary to create predictive toxicology applications. While creating an ontology (even for a rather limited domain) by consensus is a challenge by itself, the value of having multiple independent implementations of services using the ontology is enormous, as it clearly highlights components that have not been explicitly described and are thus open to diverging and possibly conflicting interpretations.

The validation and reporting services provide scientifically sound means for assessing the output of modelling work in full compliance with the REACH legislation and the OECD guidance documents. The authentication, authorisation and policy management services provide the required level of data protection management in a flexible way. The range of existing end-user applications interacting with the currently deployed OpenTox web services through the OpenTox API support the solving of real-world problems in predictive toxicology and also improve the sustainability outlook for the OpenTox Framework beyond the lifetime of the OpenTox FP7 project. The OpenTox client libraries reduce to a great extent the required effort to integrate third-party software tools with the OpenTox framework, thus encouraging the further enlargement of the OpenTox ecosystem.

The OpenTox API could be implemented equally well either as a completely new project or as an extension of an existing software. The final framework demonstrates the ability of the OpenTox API to provide a unified interface to diverse algorithms and data, and to define explicit relationships between data and processing routines. Last but not least, the services provide a sound basis for building web mashups, end-user applications with friendly GUIs, as well as embedding the functionalities in existing workflow systems.

[65] The Chemistry Development Kit http://cdk.sourceforge.net/, accessed on Apr 30, 2011

[66] Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen EL: The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. J. Chem. Inf. Comput. Sci. 43(2) :493-500.

[67] Steinbeck C, Hoppe C, Kuhn S, Floris M, Guha R, Willighagen EL: Recent Developments of the Chemistry Development Kit (CDK) - An Open-Source Java Library for Chemo- and Bioinformatics. Curr. Pharm. 2006, 12(17):2111-2120.

[68] The open source chemistry toolbox OpenBabel http://openbabel.org/, accessed on Apr 30, 2011