



Deliverable D4.3

Final version of (Q)SAR algorithms

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



Contract No.	Health-F5-2008-200787	
Document Type:	Deliverable Report	
WP/Task:	WP4	
Name	Final version of (Q)SAR algorithms	
Document ID:	OpenTox Deliverable Report 4.3	
Date:	August 31, 2011	
Status:	Final Version	
Organization:	TUM	
Contributors	Stefan Kramer Fabian Buchwald Jörg Wicker Pantelis Sopasakis Nina Jeliaskova Rama Kaalia Indira Gosh Sunil Chawla Andreas Maunz Dmitry Druzhilovsky Alexey Zakharov Barry Hardy Roman Affentranger	TUM TUM TUM NTUA AMBIT JNU-SL JNU-SL JNU-SL ALU-FR IBMC IBMC DC DC

Distribution:	Public
---------------	--------

Purpose of Document:	To document results for this deliverable
----------------------	--

Document History:	1 - First Draft v0.1 on 31 May 2011 2 - Second Draft v0.1 on 18 August 2011 3 - Updated Draft on 27 October 2011
-------------------	--

Table of Contents

Table of Contents	3
Summary	7
1 Introduction	8
2 Implementation Principles	8
2.1 Programming Languages and Libraries	8
2.2 Restful Web Service Architecture	9
2.3 OpenTox Algorithm Ontology	10
2.4 OpenTox Algorithm and Model Application Programming Interfaces (APIs)	10
3 Results	12
3.1 Previous Achievements	12
3.2 Overview on Integrated Algorithms	12
3.2.1 Fast Conditional Density Estimation (FCDE)	12
3.2.2 Instance-Based Structure-Activity Relationships (iSAR)	13
3.2.3 LoMoGraph	14
3.2.4 BBRC	15
3.2.5 LAST-PM	16
3.2.6 MaxTox	16
3.2.7 ToxTree	18
3.2.8 Applicability Domain	19
3.2.9 Generic wrapper for WEKA classification, regression and clustering algorithms	19
3.2.10 3D Structure Generation, Based on MOPAC (Molecular Orbital PACKage)	19
3.2.11 Semiempirical Quantum Chemistry Descriptors, based on MOPAC	20
3.2.12 The CDK Descriptors	20
3.2.13 pKa estimation	20
3.2.14 Site of Metabolism Estimator (SOME)	20
3.2.15 Finder	20
3.2.16 Structural Clustering	20
3.3 Workflow Management Systems	21
3.4 Graphical User Interface (GUI) for Descriptor Calculation	24
3.5 Suggested Algorithms	25
4 Algorithms Presentation	27
4.1 Descriptor Calculation Algorithms	27

4.2	Feature Selection Algorithms	27
4.3	Classification and Regression Algorithms	27
4.4	Clustering Algorithms.....	27
4.5	Applicability Domain Algorithms	27
4.6	Miscellaneous Algorithms.....	28
4.7	Algorithm Description.....	28
a.	General Information about the Service.....	28
b.	Request/Response Information.....	28
c.	Status Codes.....	28
d.	Implementation Information	28
e.	Examples	28
5	Final Algorithm Documentation.....	29
5.1	Retrieving information and applying OpenTox algorithm web services.....	29
5.1.1	Retrieving information of an algorithm.....	29
5.1.2	Applying an algorithm	30
5.2	Models for REACH relevant endpoints.....	31
5.2.1	Example TUM models	31
5.2.1.1	KNN model for Caco-2 Permeability:	31
5.2.1.2	Decision tree model for Micronucleus Data	31
5.3	Descriptor Calculation Algorithms	31
5.3.1	FreeTreeMiner.....	31
5.3.2	FMiner	33
5.3.3	gSpan'	34
5.3.4	MakeMNA	36
5.3.5	MakeQNA.....	38
5.3.6	JOELib2	40
5.3.7	OpenBabel	43
5.3.8	3D structure generation, based on MOPAC (Molecular Orbital PACKage).....	45
5.3.9	Semiempirical quantum chemistry descriptors, based on MOPAC (Molecular Orbital PACKage).....	46
5.3.10	AMBIT	47
5.3.11	The Chemistry Development Kit (CDK).....	49
5.4	Classification and Regression Algorithms	53
5.4.1	Gaussian Processes for Regression	53
5.4.2	Lazar.....	55
5.4.3	KNN	57

5.4.4	J48.....	59
5.4.5	M5P	61
5.4.6	Fuzzy-means.....	63
5.4.7	MakeSCR.....	64
5.4.8	ToxTree	66
5.4.9	WEKA machine learning algorithms	68
5.4.10	Bayes Net.....	70
5.4.11	Linear Regression	72
5.4.12	PLS.....	74
5.4.13	LoMoGraph	75
5.4.14	Interval Estimators	77
5.4.15	iSAR	79
5.4.16	Multiple Linear Regression.....	81
5.4.17	Support Vector Machine.....	83
5.4.18	Radial Basis Function Neural Network	85
5.4.19	MaxTox	88
5.5	Clustering Algorithms.....	89
5.5.1	Structural Clustering.....	89
5.6	Feature Selection, Data Transformation and Filtering Algorithms	91
5.6.1	Information Gain Attribute Evaluation.....	91
5.6.2	Chi Squared Attribute Evaluation	93
5.6.3	PCA.....	95
5.6.4	Partial Least Squares Filter	97
5.6.5	Scaling Filter	99
5.6.6	Missing Values Replacer	101
5.7	Domain of Applicability Estimation Algorithms.....	103
5.7.1	Leverage	103
5.7.2	Several descriptor-based and structure-based applicability domain algorithms.....	105
5.8	Miscellaneous Algorithms	107
5.8.1	Compound, Similarity and Substructure Search	107
5.8.2	Structure Diagram Generation, Integrated with Compound Service.....	109
5.8.3	Structure Diagram Generation by SMILES	110
6	Conclusion.....	111
7	References	111

8	Appendix A	114
8.1	A.1 gSpan output.....	114
8.2	A.2. Learn and validate a LoMoGraph model.....	118

Summary

This report on the final version of (Q)SAR Algorithms presents the work that has been accomplished within the OpenTox FP7 project on the implementation of (Q)SAR and supporting algorithms. (Q)SAR algorithm development in OpenTox has followed the RESTful Web Service architecture and all OpenTox web services are compliant with the current OpenTox Application Programming Interface (API) version 1.2 and the algorithm ontology. To be consistent with the open source philosophy of OpenTox, open source tools have been utilized for developing the initial algorithm prototypes and all source code is publicly available. We provide here a detailed overview and a comprehensive documentation of the algorithms that have been developed.

We present an overview of the implementation principles that we followed. We decided to use REST as a web service architecture since it is lightweight and easily enables the integration of existing software tools from OpenTox partners, written in different programming languages, into a common framework. We also describe briefly the OpenTox Algorithm Ontology, followed by a description of the OpenTox Algorithm and Model Application Programming Interfaces (APIs). We report on newly developed algorithms, Graphical User Interfaces (GUIs) and workflows that were devised for predictive toxicology tasks. Examples for applications regarding descriptor calculation and model learning are presented, as well as the reason for the application of certain algorithms and descriptors for modelling REACH-relevant endpoints. The rationale behind newly devised algorithms is also provided. We introduce the functional categories of the algorithms (descriptor calculation, feature selection, clustering, and model learning, which in turn can be further divided into classification and regression, estimating the domain of applicability and miscellaneous algorithms) and give an overview of the structure that is used to describe the algorithms, in particular with respect to the implementations. Generic information on how to call/apply different kinds of algorithms is presented. In addition, expected results are discussed and application scenarios of algorithms, e.g. for REACH purposes or in evaluation procedures, are presented as well as complex workflows that satisfy typical use cases. We also provide detailed information about each algorithm that has been included in the final version as of August 2011, presented in a uniform tabular format. The final OpenTox algorithms encompass:

- 11 descriptor calculation algorithms, representing both well-known existing methods such as OpenBabel, JOELib2, The Chemistry Development Kit (CDK) and gSpan', and new developments such as FMiner, MakeMNA, and MakeQNA. The single descriptor calculation web service AMBIT – developed by OpenTox partner IDEA – offers descriptors calculated by several packages;
- 19 classification and regression algorithms, ranging from decision trees (ToxTree), simple linear regression, multiple linear regression, support vector machines, fuzzy-means, nearest neighbour methods (e.g. Lazar, kNN), to complex machine learning algorithms such as Bayes Net, LoMoGraph, MaxTox and iSAR;
- 1 clustering algorithm, namely structural clustering based on frequent subgraph mining;
- 6 feature selection, data transformation and filtering algorithms of various complexity, including a partial least-squares filter, principle components analysis (PCA), chi-squared attribute evaluation, information-gain attribute evaluation, a scaling filter, and a missing value replacer;
- 2 applicability domain estimation algorithms, one of them being the leverage algorithm, the other involving several descriptor-based and structure-based applicability domain algorithms implemented in AMBIT;
- 3 miscellaneous algorithms, one of them being a similarity and substructure search algorithm for compounds, while the other two involve generation of structure diagrams either integrated with the compound web service, or based on SMILES strings.

1 Introduction

Numerous and diverse approaches for predicting toxicity via (Q)SAR have been proposed in the literature, and ongoing scientific efforts in various complementary fields have led to a large number of algorithms that are available and potentially useful for (Q)SAR and related tasks. During the first six months of the OpenTox project, a comprehensive review of available algorithms was performed by project partners considering algorithms that are common in the field of (Q)SAR, Predictive Toxicology and Machine Learning. Besides standard algorithms such as Partial Least Squares, Linear Regression or Neural Networks that are frequently used in (Q)SAR studies, in house algorithms from OpenTox partners were taken into account that have been developed and can be exposed by project participants as web services. The algorithms were grouped into 5 categories: descriptor calculation algorithms, classification and regression algorithms, clustering algorithms, feature selection algorithms and algorithms for the aggregation of results from multiple QSAR models. The OpenTox report D4.1 on Algorithm Selection and Evaluation (2009), gave a detailed description of these algorithms and, based on a set of multiple selection criteria, made a prioritization of the algorithms which indicated in which stage of the OpenTox project each algorithm was planned to be integrated into the OpenTox Framework. The OpenTox project involves many partners and developers having different programming backgrounds and experience. A common collaboration framework based on the RESTful Web Service architecture was defined for algorithm implementation that supports independent deployment of services by different partners, which may be combined in an interoperable manner into OpenTox-based applications or use cases. This framework makes it convenient for third-party (Q)SAR and machine learning researchers and developers to integrate their own tools within the OpenTox Framework through well-established HTTP methods. The implementations are arranged in a way that requests are forwarded from one server to the other, minimizing latency time and boosting performance. Additionally, all communications are stateless in the sense that none of the nodes needs to store any data; requests are self-contained and provide all the data needed by the application to process them.

In this deliverable, the final version of each (Q)SAR algorithm that is integrated into the OpenTox Framework is presented in a uniform tabular format. It gives a brief algorithm description, defines responsible partners and contact persons, provides links to the resources and the OpenTox Application Programming Interface (API) and shows examples how information about the algorithm can be retrieved or how the algorithm can be used, e.g. for descriptor calculation or model learning. More technical information (such as input parameters and output results, status codes, programming languages, dependencies like external libraries) is also included in each algorithm description. Algorithm descriptions are completed with examples that can be used to test and evaluate efficiency of the implementations.

2 Implementation Principles

Algorithms constitute a major component of the OpenTox Framework and should be in line with the key design principles of OpenTox: interoperability, extensibility and compliance with user requirements and use cases. Special care was taken to allow independent development of different algorithm software components. In the following sections the principles that were followed during algorithm implementation are presented.

2.1 Programming Languages and Libraries

Java¹ was the main programming language used by most partners, but other languages such as Ruby², and R³ were utilized as well. The implementation process also involved the utilization of widely-used and well-

¹ <http://home.java.net/>

² <http://www.ruby-lang.org/en/>

accepted open source machine learning, data mining and chemical libraries, such as WEKA⁴, OpenBabel⁵, CDK⁶ and JOELib²⁷. These tools provide a wide range of functionality in open source libraries, which makes it possible to easily include important functions in the framework.

2.2 Restful Web Service Architecture

The main reason for using a web service architecture is to combine existing software tools which are implemented in a wide range of programming languages. Web services are a popular standard for sharing data and functionality among loosely-coupled, heterogeneous systems. In particular the Representational State Transfer (REST) web service architecture⁸ was chosen because of the following advantages:

1. The produced web services are stateless;
2. The produced web services have a uniform interface (the only allowed operations are the HTTP operations GET/POST/PUT/DELETE);
3. The resources are uniquely identified by Uniform Resource Identifiers (URIs) and described by representations;
4. Components manipulate resources by exchanging representations of the resources.

All algorithm resources have representations providing information about the type of algorithm, what the algorithm accepts as input, tuning parameters, default parameter values, etc. Most algorithms and model resources in OpenTox are available in multiple representations. The Resource Description Framework (RDF) representation⁹, and in particular its XML-formatted variant, was chosen as the main data exchange format, because of the following reasons:

1. RDF is a W3C recommendation: RDF-related representations such as rdf/xml and rdf/turtle are w3c recommendations so they constitute a standard model for data exchange;
2. RDF is part of Semantic Web Policy: RDF as a representation for a self-contained description of web resources contributes to the evolution of the Semantic Web; a web where all machines can “understand” each other;
3. RDF is designed to be machine-readable: while humans can, in principle, read RDF documents, it is unlikely that they are able to understand them easily. RDF is intended to be understood by computers, not people.

Some services support additional representations like JavaScript Object Notation (JSON)¹⁰, YAML¹¹ or Application/X-Turtle¹². Some model learning services provide Predictive Model Markup Language (PMML)

³ <http://www.r-project.org>

⁴ <http://www.cs.waikato.ac.nz/ml/weka/>

⁵ <http://openbabel.sourceforge.net/>

⁶ <http://sourceforge.net/projects/cdk/>

⁷ <http://www-ra.informatik.uni-tuebingen.de/software/joelib>

⁸ <http://www.ibm.com/developerworks/webservices/library/ws-restful/>

⁹ <http://www.w3.org/RDF/>

¹⁰ <http://www.json.org/>

¹¹ <http://www.yaml.org/>

¹² <http://www.w3.org/TeamSubmission/turtle/>

representations, designed by the Data Mining Group¹³, to improve their portability, since many machine learning applications like KNIME¹⁴ and WEKA provide support for PMML.

2.3 OpenTox Algorithm Ontology

The graphical representation of the algorithm ontology used in OpenTox is shown in Figure 1. A formal OWL¹⁵ representation of the algorithm ontology is available at

<http://opentox.org/data/documents/development/RDF%20files/AlgorithmTypes/view>.

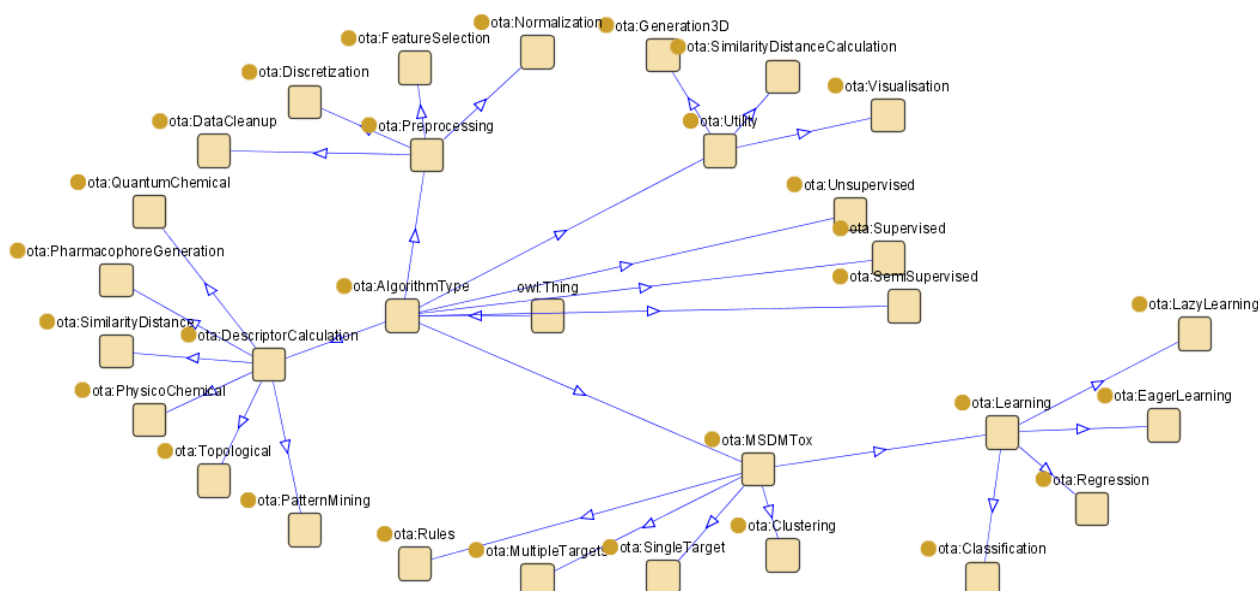


Figure 1: OpenTox Algorithm Type Ontology

In this ontology every algorithm in the OpenTox Framework is fully described, including references, parameters and default values. This is achieved by adopting the Blue Obelisk ontology¹⁶. The RDF representation of an Algorithm contains metadata described by the Dublin Core¹⁷ Specifications for modelling metadata (DC Namespace) and the OpenTox namespace. The establishment of an ontological base for the services facilitates the extension of the services and the introduction of new algorithms and new algorithm classes.

2.4 OpenTox Algorithm and Model Application Programming Interfaces (APIs)

Algorithm and Model APIs are part of the OpenTox API that enables interaction among all OpenTox software components. The current OpenTox API version is API 1.2 (<http://www.opentox.org/dev/apis/api-1.2>). In terms of REST, each algorithm and each model is represented by a resource. For example, a representation of an algorithm contains information about the input a client should provide (obligatorily or optionally) to invoke the underlying procedure (e.g. training data, prediction feature, tuning parameters, etc.). All algorithm resources

¹³ <http://www.dmg.org/>

¹⁴ <http://knime.org/>

¹⁵ <http://www.w3.org/TR/owl-features/>

¹⁶ <http://qsar.svn.sf.net/viewvc/qsar/trunk/qsar-dicts/descriptor-ontology.owl?revision=218>

¹⁷ <http://dublincore.org/>

are placed under *someDomain/algorithm* and all model resources under *someDomain/model* where *someDomain* represents a server on which the algorithm or model service is running. For example <http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/J48> is the resource of the decision tree learner J48 provided by Technische Universität München.

REST provides four basic HTTP operations (GET, POST, PUT and DELETE) that are described next.

1. GET: Return information about a resource
2. POST: Create a new instance of a resource
3. PUT: Modify a resource
4. DELETE: Delete a resource

The OpenTox algorithm API consists of two HTTP operations (GET and POST) that are described next.

1. GET /algorithm: Returns a list of all available algorithms on the server in a supported media type; e.g. text/uri-list and rdf-related media types like application/rdf+xml, text/x-triple and text/rdf+n3.
2. GET /algorithm/{id}: Returns a representation of the algorithm, identified by its *id*, in a supported media type specified in the 'Accept' header of the request.
3. POST /algorithm/{id}: A POST operation on an algorithm activates the application of the algorithm and often requires the specification of input parameters. For instance, all prediction algorithms (machine learning or otherwise) need the parameter 'dataset_uri' which is the URI of the training data set. Additionally, the parameter 'prediction_feature' is mandatory for all supervised learning algorithms; it is the target feature of the provided data set. The result from a successful POST operation is the URI of a created model.

Detailed information about the Algorithm API can be found at <http://opentox.org/dev/apis/api-1.2/Algorithm>.

The same architectural concept was applied to the construction of the model API, which provides access to all OpenTox models.

1. GET /model: Retrieve a list of all models on the server.
2. GET /model/{id}: Get the representation of a certain model in a supported media type. This representation contains mainly information about the training algorithm that produced the model, the training data set, the independent, dependent and predicted features of the model and the various training parameters (tuning parameters). An RDF/XML representation is mandatory. Additionally, some services support additional media types such as RDF N3, PMML, JSON and YAML.
3. GET /model/{id}/independent: Get the list of independent features of the model. Features are resources themselves so they are characterized by a URI. The independent features of the model are the features of the training data set, excluding the prediction feature. This feature list is available in all RDF-related media types (application/rdf+xml is mandatory) and in text/uri-list format as well.
4. GET /model/{id}/dependent: Get the dependent feature of the model that is the prediction feature of the training set that was used to learn the model. In a POST call it is specified by "prediction_feature".
5. GET /model/{id}/predicted: This is a feature that is generated along with the creation of the model. It is the feature related to the predicted values of the dependent feature using this model.
6. DELETE /model/{id}: A model can be deleted.
7. POST /model/{id}: A data set or a compound URI can be posted to a model in order to get a prediction by that model. The service exploits the underlying model to calculate the predicted values and returns

8. to the client (within the response body), a URI for the created data set. This data set contains the submitted compounds and their corresponding predictions.

More information about the model API is available at the address <http://opentox.org/dev/apis/api-1.2/Model>.

3 Results

This section gives an overview of the achievements and the progress achieved.

3.1 Previous Achievements

The design of the OpenTox framework provides several advantages. First, components could be developed independent of other parts of the framework. Most parts of the framework do not depend on others and can be developed and used on their own. Second, the components of the framework can exchange data between each other due to REST specifications. Finally, the framework itself can be easily extended with new algorithms and features. Also, many standard machine learning and (Q)SAR algorithms are already included in the framework.

3.2 Overview on Integrated Algorithms

Six OpenTox partners (IDEA, IST, TUM, NTUA, IBMC, SL-JNU) have been involved in algorithm development. Besides suggested algorithms, we integrated several new algorithms for clustering, (Q)SAR modelling, estimating the domain of applicability or visualization into the OpenTox Framework. These were not suggested in the first deliverable and were included during the last year. In the following we will briefly describe the new algorithms:

3.2.1 Fast Conditional Density Estimation (FCDE)

To quantify uncertainty in QSAR prediction, the conditional density of activity, given the structure, instead of a point estimate is used. Using a conditional density estimate, FCDE derives prediction intervals of activities. Three types of conditional density estimators are provided: histogram, normal and kernel estimators. These are based on generic machine learning algorithms. Note that in this way an arbitrary number of attributes can be used to determine a conditional density estimate. To illustrate the concept of conditional density estimation in the context of QSAR, Figure 2 shows the 2D chemical structure of a compound, its actual target value and its density distributions for the target variable obtained with three conditional density estimators: a histogram estimator, a normal estimator, and a kernel estimator (see below). The figure shows that the histogram estimator exhibits quite sharp discontinuities. In contrast, the normal estimator can only represent unimodal activity distributions and is therefore quite limited in its expressiveness. Finally, the kernel estimator is both quite flexible and capable of smoothing. More information on the three conditional density estimators can be found in Buchwald et al. [BUC10].

In the OpenTox framework the fast conditional density estimators can be used to estimate the 95% confidence interval in which a target value or endpoint of a molecule is located. To learn a conditional density estimation model the following curl call can be applied:

```
curl -i -X POST -d 'dataset_service=http://apps.ideaconsult.net:8080/ambit2/dataset' -d
'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545' -d
'prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200' -d 'estimatorType=1' -d
'baseClassifier=J48' http://opentox.informatik.tu-
muenchen.de:8080/OpenTox/algorithm/IntervalEstimator
```

In this case, the kernel estimator (estimatorType=1) is used to predict intervals which is based on the decision tree learner J48. If this model is applied to predict intervals a lower and an upper bound are obtained, e.g. see <http://apps.ideaconsult.net:8080/ambit2/dataset/657723>. Both boundaries confine the predicted 95%

interval, i.e. the model estimates with probability 95% that the true target value of a test molecule lies in this interval.

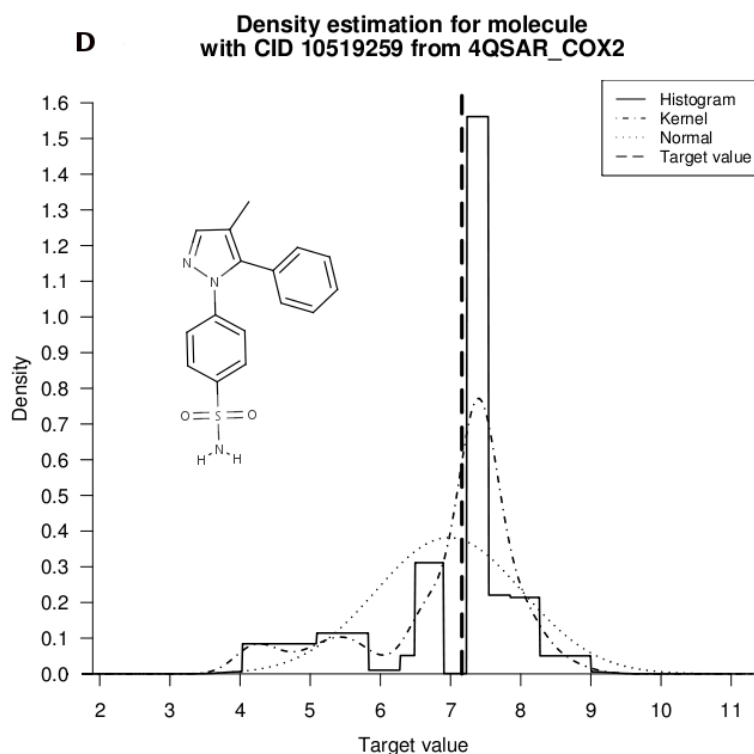


Figure 2: Example conditional density estimation with the histogram, normal and kernel estimator

3.2.2 Instance-Based Structure-Activity Relationships (iSAR)

iSAR [SOM07] consists essentially of three simple, yet effective data mining techniques for lazy structure-activity relationships (SARs) of noncongeneric compounds. In lazy SARs, classifications are particularly tailored for each test compound. Therefore, it is possible to make the most of the structure of a test compound. Substructures of the test compound are derived and used to determine similar structures. To obtain a well-balanced and representative set of structural descriptors, this set is enriched by strongly activating or deactivating fragments from the training set and subsequently redundant fragments are removed. Finally, k -Nearest Neighbour classification for several values of k is performed and it is voted among the resulting predictions. These techniques (enrichment, removing redundancy, and voting) are integrated into the system iSAR. Experiments on three datasets indicated that this simple and lightweight approach performs at least on the same level as other, more complex approaches.

Figure 3 shows a data flow of feature generation and selection in iSAR: All paths occurring in a test instance and the basic feature set are selected along with additionally highly significant substructures not occurring in the test instance. The final feature set consists of the closed features within this selected set.

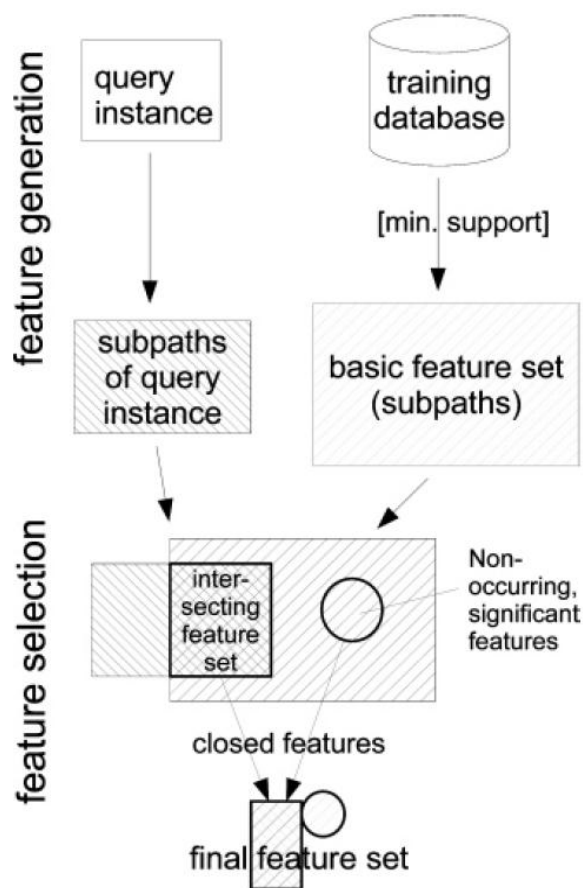


Figure 3: Data flow of feature generation and selection in iSAR

3.2.3 LoMoGraph

Local model learning for graph classification and regression (LoMoGraph) [BUC11] detects groups of structures for local (Q)SAR modelling. The algorithm combines clustering and classification or regression for making predictions on chemical structure data. A clustering procedure producing clusters with shared structural scaffolds is applied as a preprocessing step, before a (local) model is learned for each relevant cluster. Instead of using only one global model (classical approach), LoMoGraph uses weighted local models for predictions of query compounds dependent on cluster memberships. Thus, LoMoGraph is an interplay of structural clustering, model learning and prediction. A graphical overview of LoMoGraph is shown in Figure 4. From Figure 4 two important characteristics of the structural clustering procedure can be seen. It is overlapping and non-exhaustive, i.e. a molecule can fall into no cluster, one cluster or multiple clusters. If it falls into no cluster, the global model is applied for prediction. It can be seen as a default or backup model. If the query molecule falls into a single cluster, the local model based on this cluster is used for prediction, and if it is assigned to multiple clusters, all relevant local models are used for prediction, where different weights dependent on the cluster size are assigned.

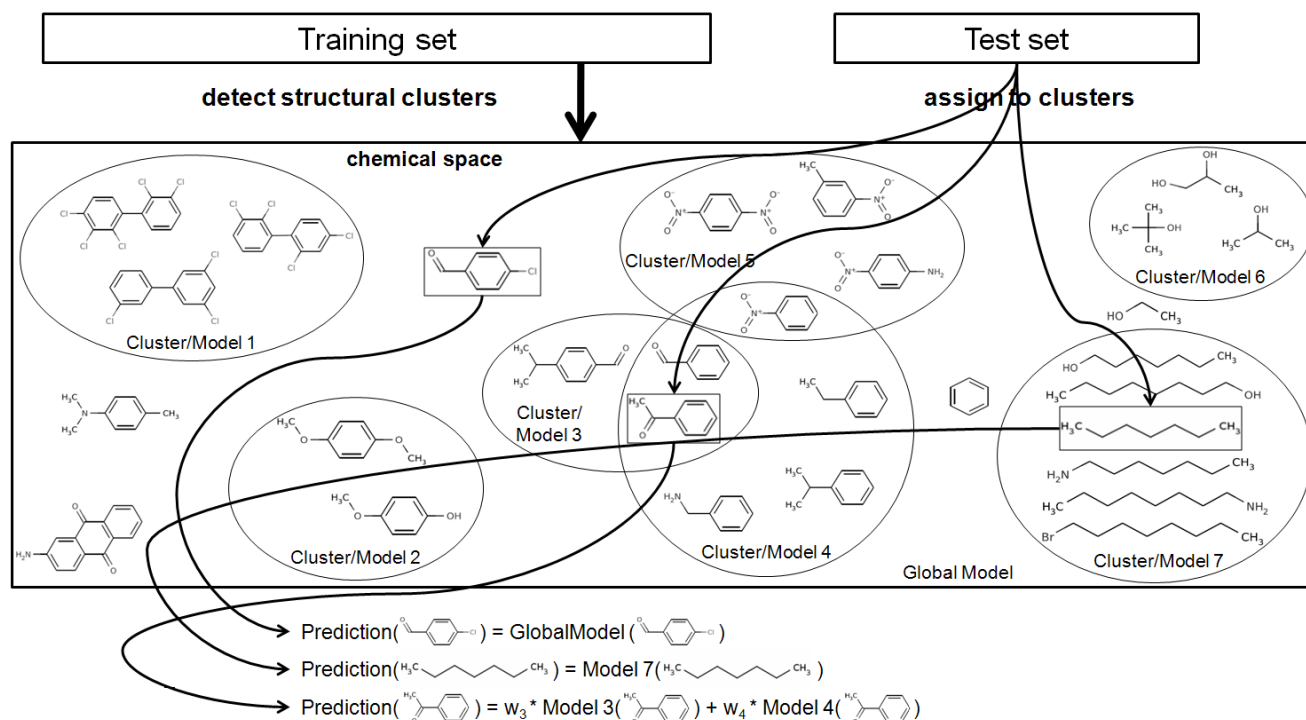


Figure 4: Graphical overview of LoMoGraph

LoMoGraph as provided in the OpenTox framework can be used, e.g., for modelling REACH relevant endpoints. For predicting Caco-2 Permeability the following LoMoGraph model can be used: http://opentox-dev.informatik.tu-muenchen.de:8080/OpenTox-dev/model/TUMOpenToxModel_LoMoGraph_3. This model consists of one global and several local models. During model learning, first a structural clustering procedure is applied with a structural overlap of 40%, i.e. molecules must share at least 40% structural overlap to be members of the same cluster. Then for (local) modelling, only clusters with at least 20 molecules are taken into account. For each relevant cluster a linear regression model is learned, with WEKA's standard parameter setting. To evaluate this model a 10 fold cross validation was used which resulted in, e.g., a mean absolute error of 0.421. Further statistics can be found on <http://opentox.informatik.uni-freiburg.de/validation/crossvalidation/355/statistics>. Applied curl calls that were used for modelling and validation can be found in Appendix A at the end of this document.

3.2.4 BBRC

In computational chemistry, Frequent Subgraph Mining has been widely applied to databases of pharmacological compounds to identify functional groups for drug design or hazard detection. However, the result set is typically much too large to be of actual use to most statistical learners, let alone human experts. Moreover, many very similar fragments are retrieved this way.

Backbone Refinement Class Mining (BBRC) [MAU09, MAU11] reduces the result set of substructures by structural compression and correlation to the endpoint under investigation. The method has high compression potential and handles large chemical data sets very effectively. It partitions the complete search space of subgraphs (chemical fragments) by maintaining a structural invariant (the backbone) and selects just one representative (the most significant one) from inside each partition. This modified hypothesis space has been shown to yield a robust collection of structurally diverse descriptors which cover the compounds very well. For example, data sets with more than 20,000 compounds can be processed in just a few minutes using this approach. It brings tremendous speed up in computation, very high compression while retaining good

coverage of the database, and predictive accuracy using the obtained descriptors on par with the complete set of subgraphs.

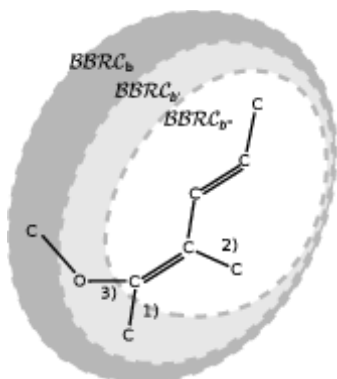


Figure 5: Backbone Refinement Class search space induced by a 2D-graph pattern

3.2.5 LAST-PM

Similar to BBRC, LAST-PM [MAU10] addresses the problem of frequent subgraph mining returning overly large results sets with many very similar fragments.

Latent Structure Pattern Mining (LAST-PM) extracts latent (hidden) information from the set of frequent and correlated subgraphs, thereby reducing the result set, incorporating correlation measures. LAST-PM, as BBRC, yields very high compression while retaining good coverage of the database. In contrast to other graph analysis methods however, LAST-PM is designed to be integrated into the actual graph mining step, not as a separate post-processing step, making it much more efficient. The engine produces elaborate patterns, integrating structural ambiguities of various sizes into the patterns, which are found by aligning and stacking subgraphs, followed by a spectral analysis step applied to this weighted graph. The resulting descriptors have been used in computational models for complex biological endpoints (bioavailability, blood-brain-barrier) and have been compared favourably to or could improve on highly optimized physico-chemical descriptors. Many of the descriptors are readily interpretable by experts.

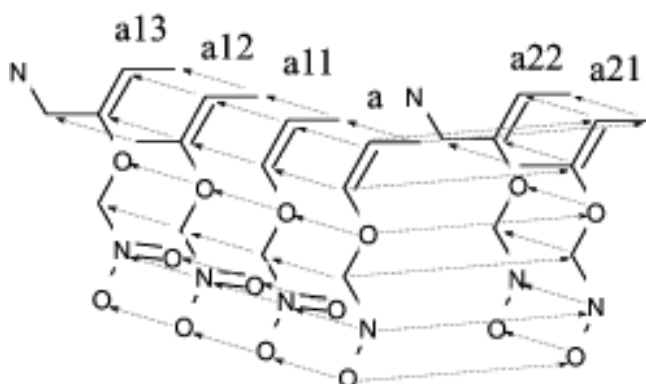


Figure 6: Latent Structure Pattern Mining search space aligning several 2D-graph patterns

3.2.6 MaxTox

MaxTox is a novel QSAR algorithm designed to predict toxicity of new compounds based on their similarity to compounds with known toxicities and having the same end-points. MaxTox uses a Maximum Common Substructure Search (MCSS) approach to predict the toxicity of an unknown test compound. These molecules are pairwise compared with each other to generate a list of Maximum Common Substructures found between

each pair. Substructures occurring in more than one compound and which consist of more than two atoms are extracted. All such generated substructures (across all pairwise comparisons) are parsed to remove redundant entries. This set of unique substructures creates a dictionary of toxic MCSS which is used to generate fingerprints of a training set of molecules. Random Forest/SVM models are built and optimized using the fingerprints as descriptors with the toxic endpoint as target value. These models are then used to predict the toxicity of new molecules as shown in Figure 7. MaxTox thus attempts to determine the relationship between the toxicity of a molecule and the MCSS it shares with other toxic molecules. The MaxTox web application (see Figure 8) provides a user friendly GUI in which the MaxTox algorithm can be used to generate descriptors (fingerprints) using the MCS dictionary. Also predictive models can be built using these descriptors and a machine learning algorithm like SVM. Alternatively, the user can select one model and use it for *in silico* prediction. The already built models using the MaxTox algorithm are available at <http://202.141.146.74:8080/MaxtoxMCSS/model>. There are Fingerprinting models that can be used to generate descriptors and Classification models that can be used to predict toxicity. For examples on how to use MaxTox we refer the reader to the MaxTox user guide available at <http://www.opentox.org/tutorials/maxtox>. Using the Predict Dataset service of MaxTox (<http://202.141.146.74:8080/MaxtoxMCSS/predict>), the user can select a model to be used for prediction, the classification model built using Bursi Mutagenicity Dataset model which makes prediction for the endpoint mutagen/non-mutagen. When the unknown dataset URI is entered in the required field and the user submits the dataset for prediction, MaxTox returns the result URI which links to the prediction results.

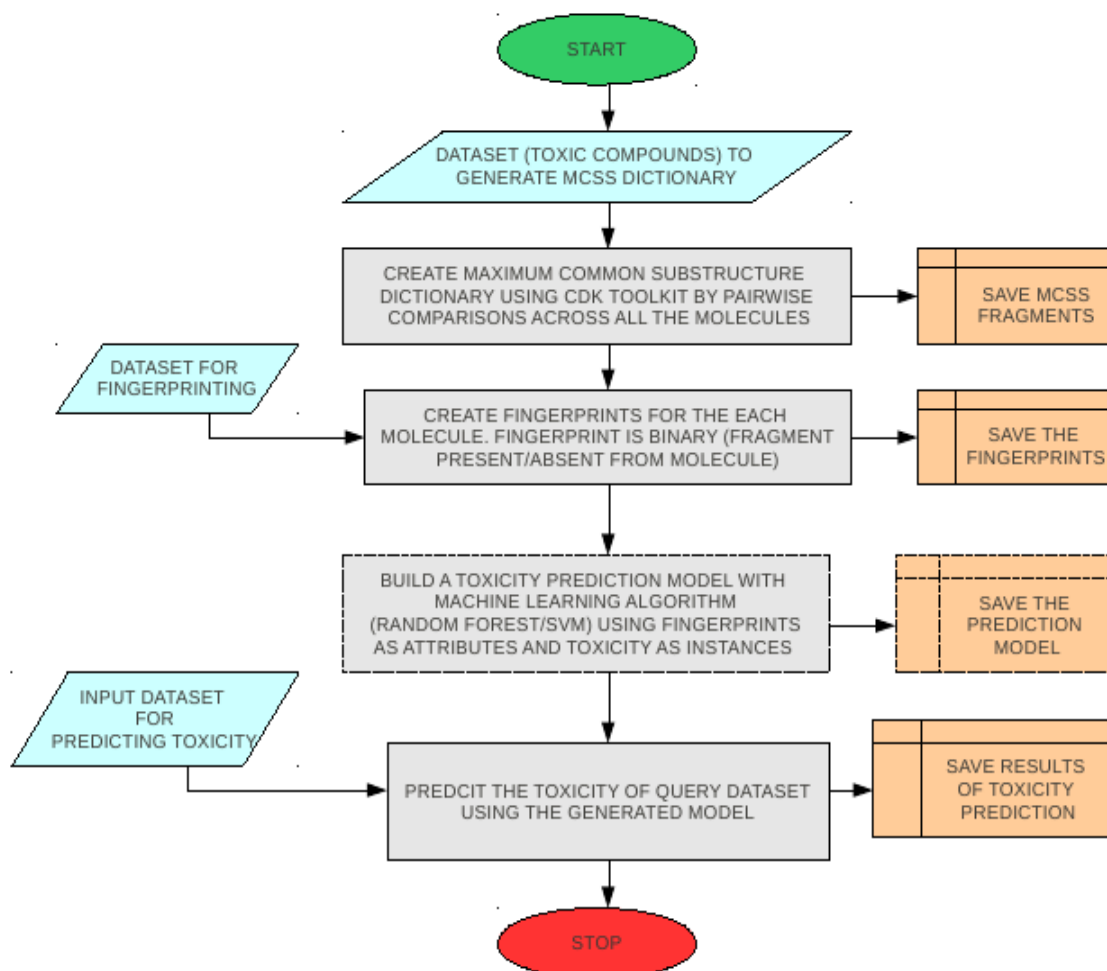


Figure 7: Flowchart for MaxTox Prediction Model building and Toxicity prediction

developed by JNU in collaboration with SL as a part of the OpenTox project.



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.
- The predictive models are then used to predict new and un-known compounds.

Maxtox is delivered as an open source application. It uses open source packages like CDK, Restlet, Jena & R to generate models and predict toxicities. It is part of the [OpenTox](#) initiative to create freely accessible resources to predict toxicity.

Maxtox can consume and respond in RDF datatypes and works with the [OpenTox API](#) specification. The Maxtox application can be used as a component of other prediction use-cases hosted from other servers, as long as the data transactions are

Figure 8: Screenshot of web application of MaxTox

3.2.7 ToxTree

The ToxTree algorithm web service is a wrapper of the ToxTree application¹⁸, which estimates toxic hazard by applying a decision tree approach. Currently, it includes the following modules. Each module is accessible as a separate algorithm service:

- Cramer rules
- Extended Cramer rules
- Verhaar scheme for predicting toxicity mode of actions
- A decision tree for estimating skin irritation and corrosion potential
- A decision tree for estimating eye irritation and corrosion potential

¹⁸ <http://toxtree.sourceforge.net/>

- f. A decision tree for estimating carcinogenicity and mutagenicity
- g. Structure Alerts for the *in vivo* micronucleus assay in rodents
- h. START biodegradation and persistence plug-in
- i. Skin sensitisation alerts
- j. SMARTCyp [RYD2010] plugin

3.2.8 Applicability Domain

The implementation of applicability domain (AD) is based on AMBIT open source code, available at <http://ambit.sourceforge.net> and used in previous applicability domain work [NET2005] [JAW2007][JAW2005]. All available AD algorithms from IDEA can be found at:

<http://apps.ideaconsult.net:8080/ambit2/algorithm?type=AppDomain>.

The algorithm AD service is used to create an AD model, specific for the training set. The model then becomes available as an OpenTox Model service. The model can be used to estimate AD by using HTTP POST operation with a `dataset_uri` parameter, pointing to the data set to be estimated. The result of the POST operation is a URL of a new data set, containing the result. To calculate the applicability domain, the following options are available that can also be found on the web

<http://apps.ideaconsult.net:8080/ambit2/algorithm?type=AppDomain>:

- k. Leverage [NET2005]
- l. Descriptor ranges in transformed PCA space [NET2005] [JAW2007][JAW2005]
- m. Euclidean distance [NET2005][JAW2007][JAW2005]
- n. City block distance [NET2005] [JAW2007][JAW2005]
- o. Mahalanobis distance¹⁹ [NET2005] [JAW2007][JAW2005]
- p. Non parametric density estimation [NET2005] [JAW2007][JAW2005]
- q. Hashed fingerprints, Tanimoto distance [JAW2007]
- r. Hashed fingerprints, number of missing bits [JAW2007]

More information on the descriptor and structural based applicability domain algorithms is provided in section 5.2.2.

3.2.9 Generic wrapper for WEKA classification, regression and clustering algorithms

Initially, linear regression, J48 decision tree and *k*-means clustering have been added as representative for the three categories. Recently, the services were extended with several more WEKA algorithms from all three classes.

3.2.10 3D Structure Generation, Based on MOPAC (Molecular Orbital PACKage)

We implemented a 3D-structure generation wrapper for the well-known semiempirical quantum chemical software Molecular Orbital PACKage (MOPAC). Given a dataset with 2D structures, it generates an optimized 3D structure, and it replaces the original structures. Given a dataset with 3D structures, it optimizes the structures and replaces the original structures.

¹⁹

http://www.aiaccess.net/English/Glossaries/GlosMod/e_gm_mahalanobis.htm

3.2.11 Semiempirical Quantum Chemistry Descriptors, based on MOPAC

We implemented a wrapper for the well-known semiempirical quantum chemical software Molecular Orbital PACKage (MOPAC) to generate quantum chemistry descriptors. Given a dataset with 2D structures, it calculates electronic descriptors NO. OF FILLED LEVELS, TOTAL ENERGY, FINAL HEAT OF FORMATION, IONIZATION POTENTIAL ELECTRONIC ENERGY, CORE-CORE REPULSION, MOLECULAR WEIGHT, EHOMO, ELUMO, by running MOPAC. The algorithm does not attempt to optimize the structure(s) and fails if no 3D structure is available.

3.2.12 The CDK Descriptors

Each descriptor is available as a separate algorithm service. The service is designed as a generic wrapper for classes, implementing the CDK IMolecularDescriptor interface and allows easy extension and adding new descriptors.

3.2.13 pKa estimation

This pKa estimation algorithm is implemented as a decision tree based on a set of SMARTS strings, as reported in [LEE2008]

3.2.14 Site of Metabolism Estimator (SOME)

SOME [ZHE2009] relies on semi-empirical quantum chemical calculations and machine learning methods for CYP450-mediated SOM prediction of six important metabolic reactions (aliphatic C-hydroxylation, aromatic C-hydroxylation, N-dealkylation, O-dealkylation, N-oxidation and S-oxidation) for eleven CYP450 isoforms. The algorithm service `/ambit2/algorithm/ambit2.some.DescriptorSOMEShell` is a wrapper of the executable²⁰. It accepts a `dataset_uri` as input parameter, and generates features. A comparison between SOME, SmartCyp and commercial software was presented as a poster at QSAR2010 [JEL2010].

<http://apps.ideaconsult.net:8080/ambit2/algorithm/ambit2.some.DescriptorSOMEShell>

3.2.15 Finder

Finder is an algorithm service, wrapping several methods of retrieving a chemical structure, given an identifier (registry number or chemical name). It accepts a data set URI as input parameter, and a `feature_uris[]` parameter, specifying the data set column, containing the identifier. The `search` parameter should be set to one of the several remote services (CIR, ChemIDPlus, PubChem, ChEBI, ChEMBL or OpenTox), which will be contacted in order to retrieve the structure. In addition, the NAME2STRUCTURE option uses an embedded instance of OPSIN[LOW2011] to generate a structure from chemical name. The `mode` parameter specifies how the resulting structure should be stored. The service is mainly intended to be used to retrieve/generate structures, when the uploaded data sets contain only chemical identifiers:

<http://apps.ideaconsult.net:8080/ambit2/algorithm/finder>

3.2.16 Structural Clustering

This method [SEE10] works on structural graph data, without generating features or decomposing graphs into parts. In contrast to many related approaches, the method does not rely on computationally expensive maximum common subgraph (MCS) operations or variants thereof, but on frequent subgraph mining. More specifically, the problem formulation takes advantage of the frequent subgraph miner gSpan (that performs well on many practical problems) without effectively generating thousands of subgraphs in the process. In the proposed clustering approach, clusters encompass all graphs that share a sufficiently large common subgraph.

²⁰ http://www.dddc.ac.cn/adme/myzheng/SOME_1_3.tar.gz

The size of the common subgraph of a graph in a cluster has to take at least a user-specified fraction of its overall size. The structural clustering procedure works in an online mode (processing one structure after the other) and produces overlapping (non-disjoint) and non-exhaustive clusters. In a series of experiments, the effectiveness and efficiency of the structural clustering algorithm on various real world data sets of molecular graphs was shown. For more information, we refer the reader to [SEE10].

During the clustering process, only connected subgraphs are considered as common subgraphs. The similarity between graphs is defined with respect to some user-defined size threshold. The threshold is set such that the common subgraphs shared among a query graph and all cluster instances make up a specific proportion of the size of each graph. A graph is assigned to a cluster provided that there exists at least one such common subgraph whose size is equal or larger than the threshold. In this way, an object can simultaneously belong to multiple clusters (overlapping clustering) if the size of at least one common subgraph with these clusters is equal or bigger than the threshold. If an object does not share a common sub-graph with any cluster that meets the threshold, this object is not included in any cluster (non-exhaustive clustering). A graphical overview is shown in Figure 9. For one graph after the other, it is decided whether it belongs to an existing cluster or whether a new cluster is created.

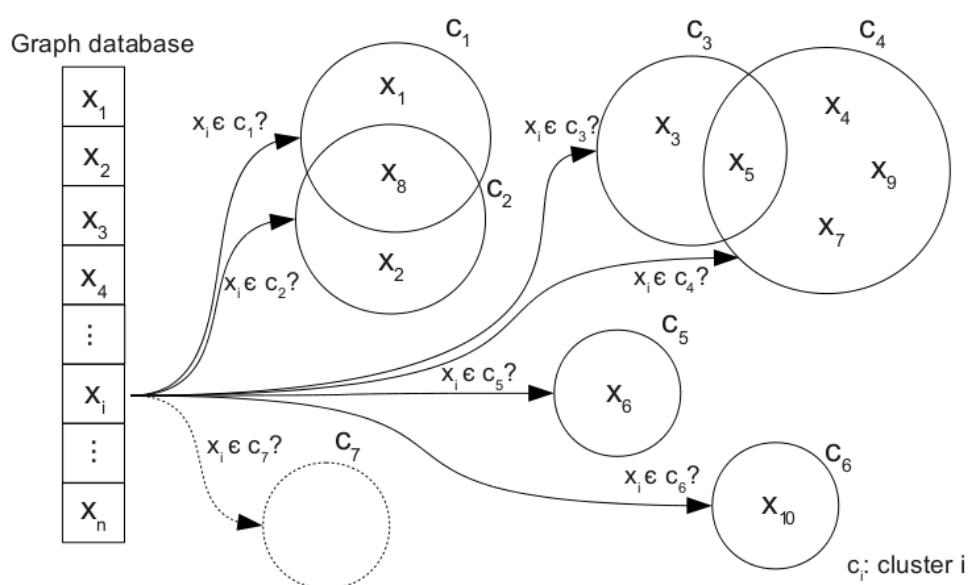


Figure 9: Schematic overview of the cluster membership assignment for instance x_i . Graph instances are represented by x_1, \dots, x_n , clusters by C_1, \dots, C_k .

3.3 Workflow Management Systems

Due to the web service based design of OpenTox, the web service and thus, the framework can be easily included in workflow management systems. Workflow management systems provide a way to combine single services to one complete workflow which can be stored and executed multiple times with different parameters. We included OpenTox services in the Taverna²¹ workbench. The workflows are shared over the myExperiment²² community and therefore can be easily included in Taverna.

²¹<http://www.taverna.org.uk>

²²<http://www.myexperiment.org/>

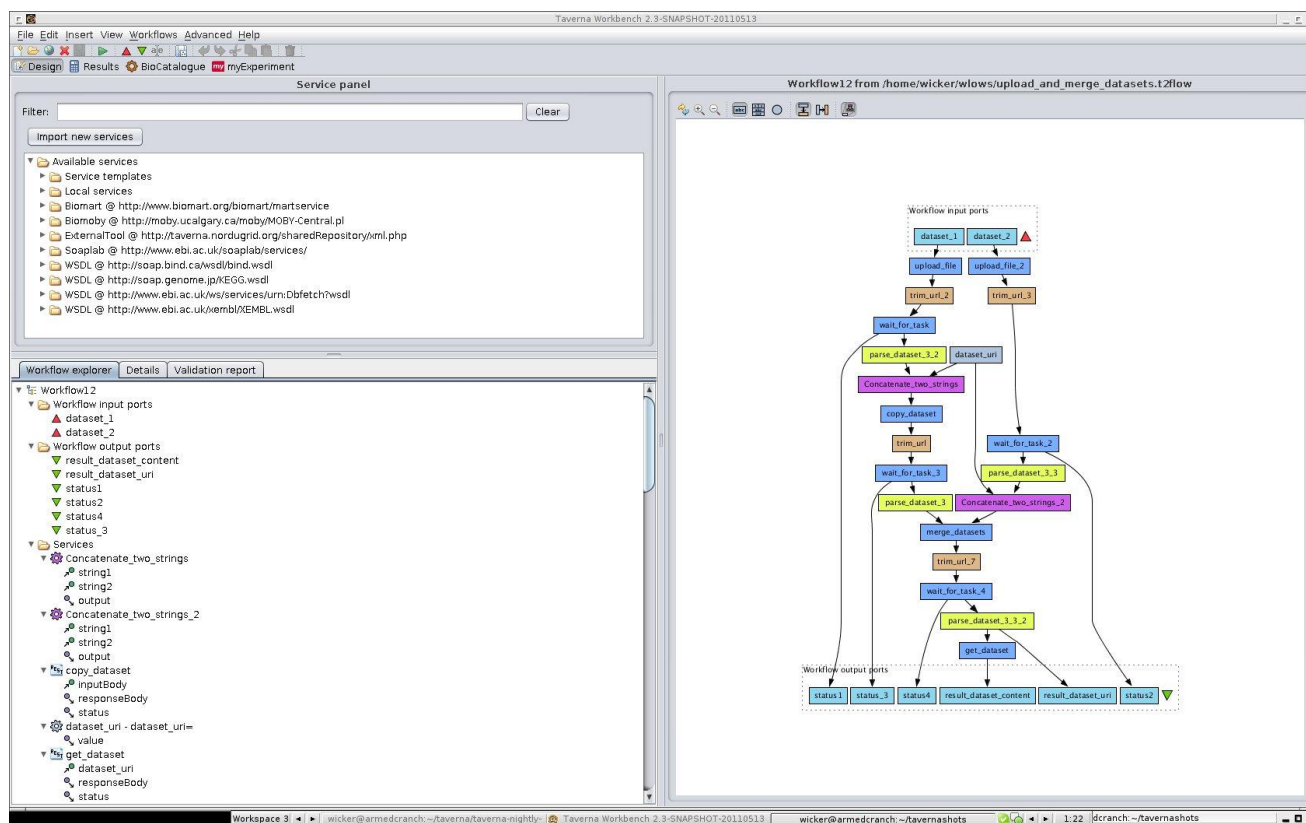


Figure 10: Example of a Taverna workflow

The latest versions of Taverna (2.3 and newer) include a REST interface. REST services can be configured and included in workflows by setting the operation (POST, PUT, DELETE or GET) and the URI of the service. The RDF output of OpenTox services can be parsed using the XPath utility included in Taverna. Using this, the output of OpenTox services can be used as input into other workflow elements. Taverna supports loops, therefore it is possible to wait for OpenTox tasks. Conditions can be included using BeanShell scripts. Nevertheless, most operations in Taverna can be performed using drag and drop or by just using the mouse. Thus, it is rather simple to implement complex workflows using OpenTox services in Taverna. Another feature gained by using Taverna is getting access to biological services. Taverna provides access to a large number of biological Web Services (see <http://www.biocatalogue.org/>).

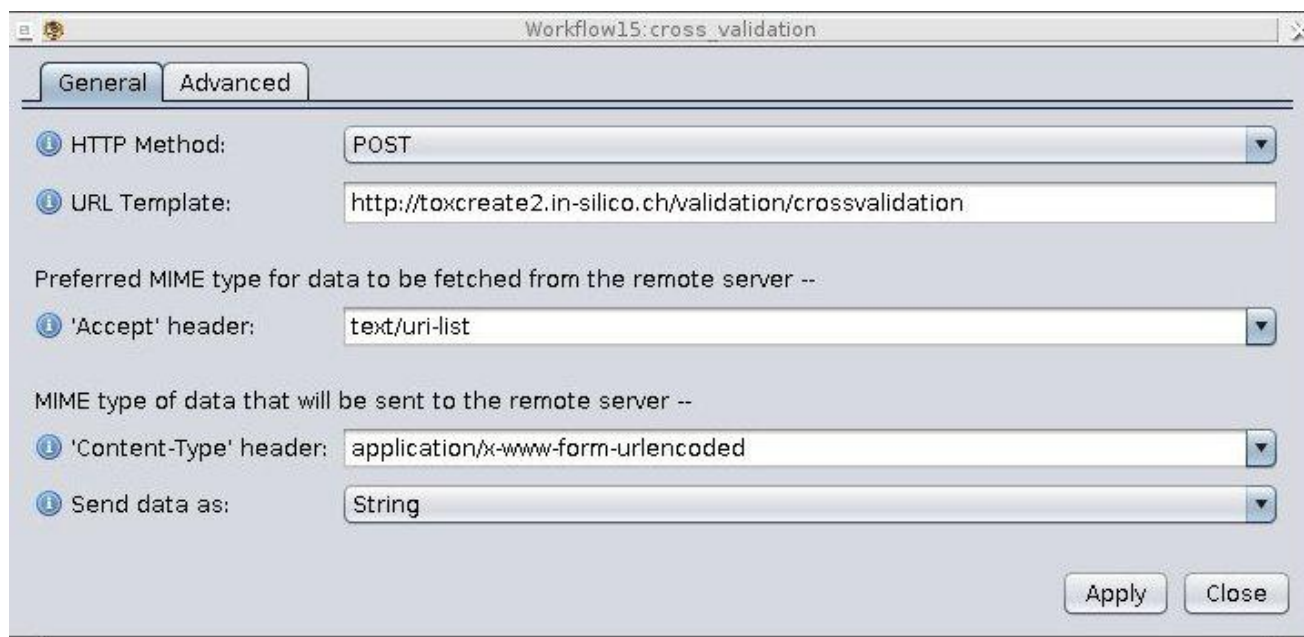


Figure 11: Configuration of a REST service in Taverna

Another example workflow is given in Figure 12. This workflow comprises

1. Feature calculation for train and test set
2. Model learning and validation of a LoMoGraph model
3. Application of the learned model to a test set

The workflow can be divided into 4 parts (see Figure 12). In the black part, CDK descriptors are calculated for both training and test set. In the red part, LoMoGraph is validated using an OpenTox validation server. In the green part, a LoMoGraph model is learned on the training set. Finally, in the blue part, the learned model is applied on the test set.

- Blue boxes are web services, e.g. LoMoGraph
- Brown boxes are BeanShell scripts, e.g. for trimming URIs (deleting newlines)
- Grey boxes contain strings, e.g. input parameters for web services
- Lilac boxes concatenate string

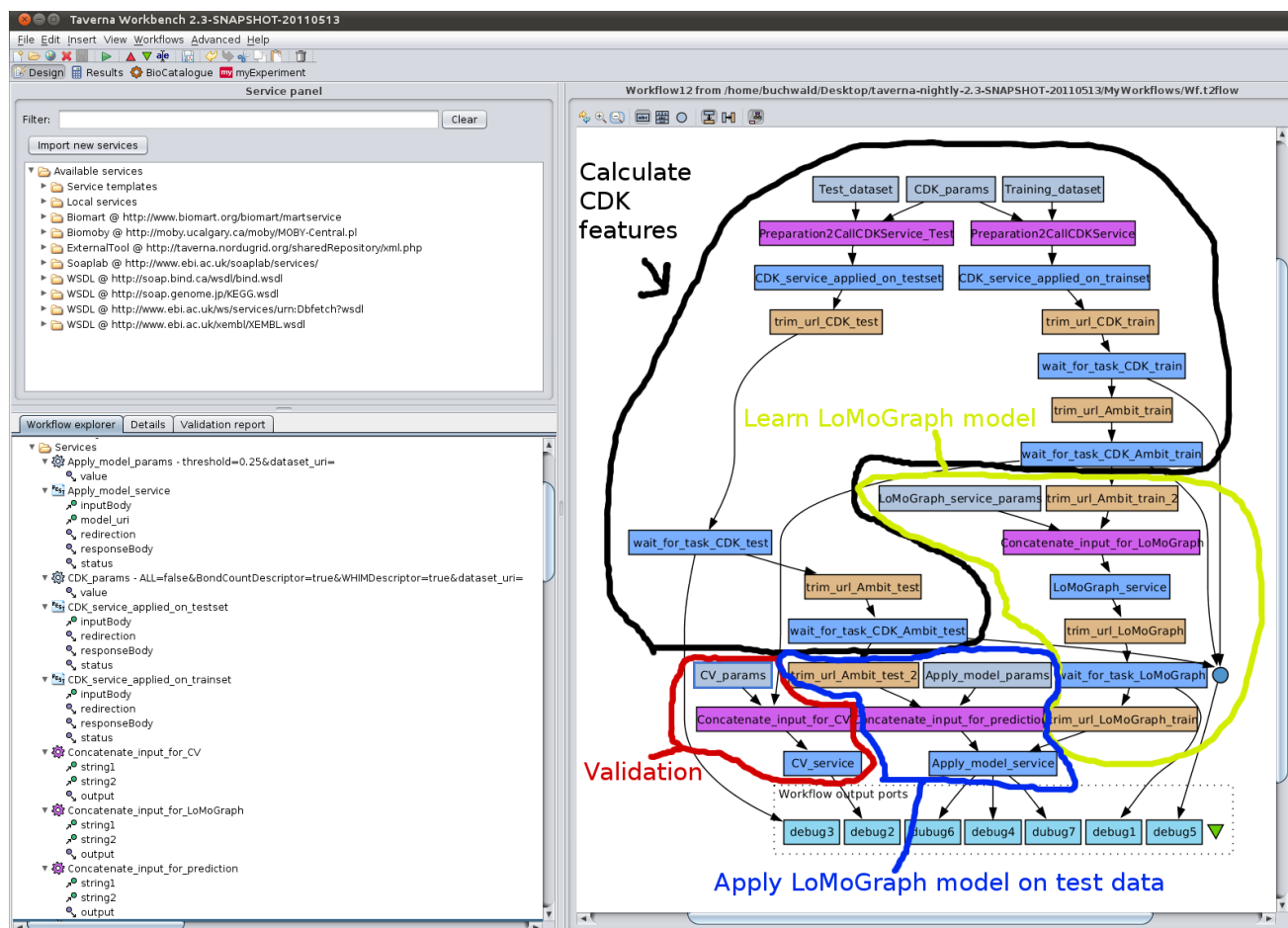
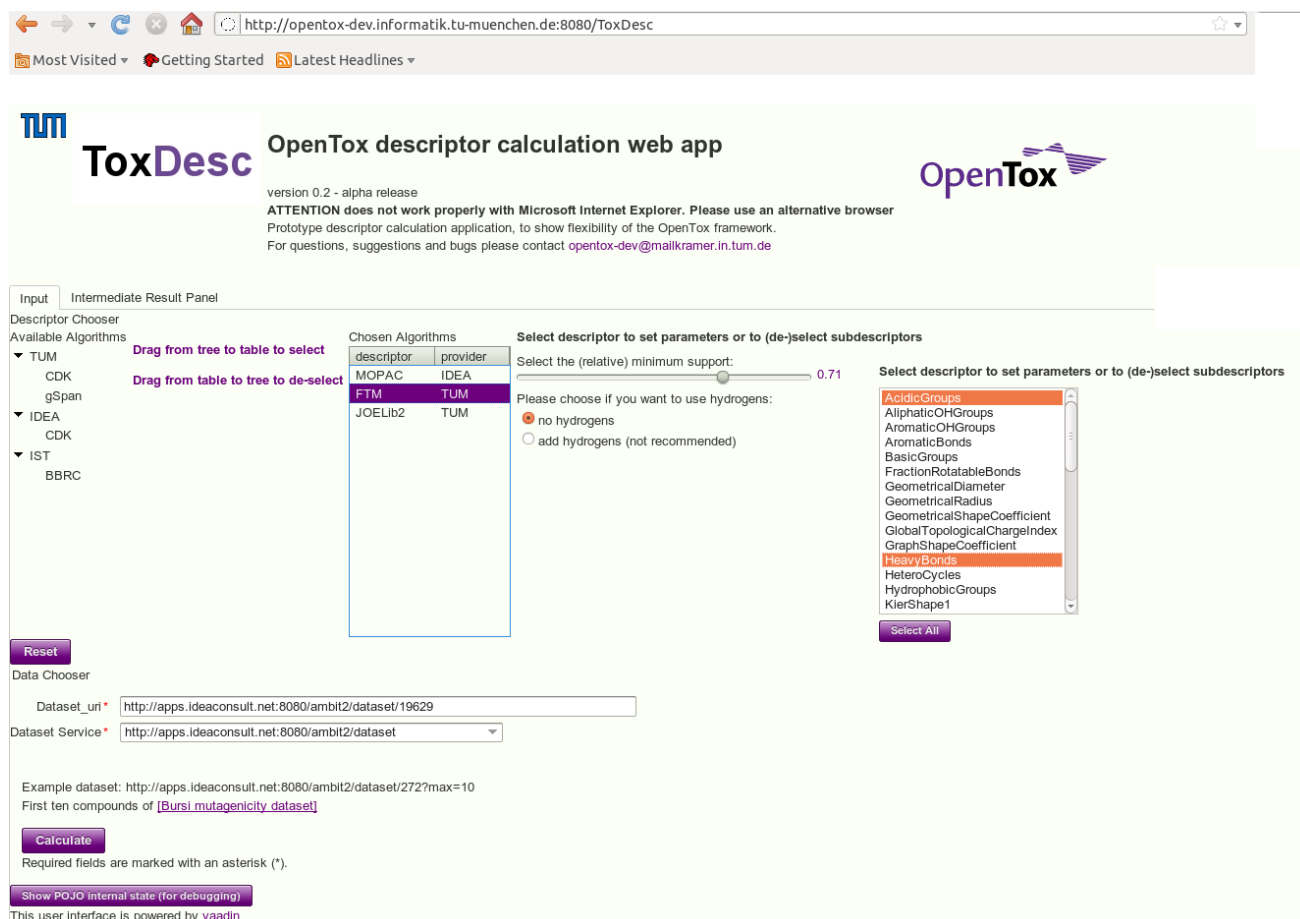


Figure 12: LoMoGraph workflow in Taverna

3.4 Graphical User Interface (GUI) for Descriptor Calculation

Besides the two graphical user interfaces (GUIs) ToxPredict and ToxCreate, to apply existing models or to build and/or validate models, we developed a third GUI, ToxDesc. It can be found at <http://opentox-dev.informatik.tu-muenchen.de:8080/ToxDesc> and used to calculate descriptors in a user-friendly way.

Figure 13 shows a screenshot from the GUI "ToxDesc". TUM's as well as IDEA's descriptor calculation algorithms can be used within it. It is possible to drag and drop the desired algorithm from the panel left to "Chosen Algorithms". Then, by clicking on a selected algorithm it is possible to set the parameters that should be calculated, e.g. Acidic Groups and Heavy Bonds for JOELib2. Additionally, a button is provided to select all descriptors. If parameters are optional like for the Free Tree Miner (FTM) where the user can choose between the option "no hydrogens" and "add hydrogens" radio buttons are used. The minimum support, i.e. the number of molecules in which a free tree must occur to be frequent, can be varied via a bar.



TUM **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@maikramer.in.tum.de

Input Intermediate Result Panel
 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)

Dataset_uri *
 Dataset Service *
 Example dataset: <http://apps.ideaconsult.net:8080/ambit2/dataset/272?max=10>
 First ten compounds of [\[Bursi mutagenicity dataset\]](#)
 Calculate
 Required fields are marked with an asterisk (*).
 Show POJO internal state (for debugging)

This user interface is powered by [yaadin](#)

Figure 13: Screenshot from ToxDesc

To run a descriptor calculation a data set URI on which the descriptors should be calculated and a data set service must be provided on which the new data set containing the calculated descriptors is stored. Finally, to start the calculation the “Calculate” button must be used. However, if a user wants to discard the parameter choice, the “Reset” button can be used to discard all settings.

3.5 Suggested Algorithms

In the first OpenTox D4.1 report on algorithms, we suggested several algorithms to be included in the OpenTox framework. Algorithms were classified into three categories A, B and C, essentially reflecting an assessment of the order according to which implementation should proceed. From the most important category A, all suggested algorithms were included at end of month 33 of the project. Whereas most of the algorithms initially rated as category B were implemented and offered as services, technical considerations (concerning the ease of integration, for instance, of proprietary Matlab toolboxes) and user requirements led us to implement a few category C algorithms before finishing all category B algorithms. For instance, MOPAC and AMBIT descriptor calculation, Gaussian Processes regression and consensus modelling, are already available, whereas the implementation of yet another method for feature selection was re-prioritized for future releases. As a highlight of algorithm development and provision, we consider, amongst others, various practical new algorithms for descriptor calculation (BBRCs, LAST-PM), structural clustering (PSCG), local modelling for QSAR (LoMoGraph) and fast conditional density estimation (FCDE) as already discussed above.

Algorithm category	Priority	Algorithm
Descriptor calculation	A	FTM (TUM) ✓ OpenBabel ✓ MakeMNA (IBMC) ✓
	B	FMiner (IST) ✓ gSpan'(TUM) ✓ MakeQNA (IBMC) ✓ JOELib ✓ CDK ✓
	C	MOPAC ✓ AMBIT ✓
Classification and regression	A	MLR ✓ kNN ✓ J48 ✓ PLS ✓
	B	SVM ✓ Lazar (IST) ✓ SMIREP/SMIPPER (ALU-FR) ToxTree (IDEA) ✓ Fuzzy-means (NTUA) ✓ MakeSCR (IBMC)
	C	Gaussian Processes for Regression ✓ iSAR (TUM) ✓ RUMBLE (TUM) M5P ✓ MaxTox (SIT-JNU) ✓
Feature selection	A	InfoGainAttributeEval ✓
	B	FCBF PCA ✓ Chi Square Feature Evaluation ✓ CFS Feature Set Evaluation
	C	Wrapper Feature Set Evaluation
Algorithms for the aggregation of results from multiple QSAR models	C	Consensus models ✓

4 Algorithms Presentation

Algorithm web services are key components in the OpenTox framework. They are used to accomplish different user scenarios (use cases) and are responsible for data manipulation, descriptor calculation, descriptor selection, reduction of dimensionality, and most importantly generation of regression and classification (QSAR) models.

Algorithms that are included in the final version of the OpenTox framework are summarized in the next sections categorized into five groups: descriptor calculation algorithms, feature selection algorithms, clustering algorithms, classification and regression algorithms and algorithms for the aggregation of results from multiple QSAR models.

4.1 Descriptor Calculation Algorithms

This category currently includes algorithms which calculate descriptors that represent chemical structures. There are two different types of molecular descriptors, namely physico-chemical and (sub-)structural descriptors. In the group of structural and sub-structural descriptors, five algorithms have been implemented (FreeTreeMiner, Fminer, gSpan, MakeMNA, MakeQNA). Also included are three sets of descriptor calculation algorithms that belong to the group of physico-chemical descriptors, namely the Chemistry Development Toolkit (CDK), JOELib2, OpenBabel and DRAGON.

4.2 Feature Selection Algorithms

This category contains algorithms for the reduction of the dimensionality of a dataset, by selecting only a subset of a full set of descriptors included in the dataset. In the OpenTox Framework InformationGainAttribute Evaluation, Chi-Squared Feature Evaluation, and PCA are provided.

4.3 Classification and Regression Algorithms

These services are responsible for the generation of (QSAR) models which are stored on the server side and can be used for predicting toxicological activity or other property values. In the final version of the OpenTox Framework we provide several common (QSAR) modelling algorithms (Linear Regression, Multiple Linear Regression, PLS, Gaussian Processes, Neural Networks), standard machine learning methods (SVMs, KNN, M5', J48, Naïve Bayes) and in house algorithms from partners (Iazar, ToxTree, BBRC, LastPM, LoMoGraph, MaxTox, iSAR, Three Conditional Density Estimators, MakeSCR) that were partly developed during the last three years.

4.4 Clustering Algorithms

This category contains unsupervised learning algorithms that group objects of similar kind into respective categories. In other words, clustering algorithms are exploratory data analysis tools which aim at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Besides standard clustering algorithms like k -means, we provide with TUM's StructuralClustering procedure an algorithm that was developed recently and is particularly suited for chemical data sets.

4.5 Applicability Domain Algorithms

The Applicability Domain (AD) of a (QSAR) represents the physico-chemical, structural or biological space on which the training set of the model has been developed, and for which it is applicable to make predictions for unseen compounds. The purpose of AD is to state whether the model's assumptions are met. In OpenTox we provide several AD algorithms that are mostly based on distances, like the Manhattan or Mahalanobis distance. Also, hashed fingerprints combined with the Tanimoto similarity coefficient as well as the Leverage method

(AMBIT and NTUA implementations) are provided. The usage of AD follows the OpenTox API for algorithms. More information on this topic can be found in the OpenTox D2.2 report on the Prototype Framework.

4.6 Miscellaneous Algorithms

We provide several miscellaneous algorithms that can be used for substructure search or visualization.

4.7 Algorithm Description

It should be noted that all algorithms mentioned above follow the OpenTox APIs, regardless of the category to which they belong. Each algorithm is described in the next section of this report by a short text description and a detailed description of the web services that have been developed to implement the algorithm. Each algorithm implementation is presented in a uniform tabular format that has five logical parts, described below:

a. General Information about the Service

The first part of the table provides a description of the service, accompanied by the URI of the resource and a reference to the current API. "non-AA" and "AA" indicates whether the service supports Authentication & Authorization. Other fields of this table indicate the partner who is responsible for the service and a contact person who can provide more information about the implementation. Additionally, algorithm-specific information is provided such as type of descriptors for descriptor calculation algorithms or whether a modelling algorithm can be used for regression or classification.

b. Request/Response Information

All algorithm web services require a number of input parameters and respond by providing the URI of the produced resource (model URI, dataset URI or feature URI). Input parameters can be mandatory or optional. If they are optional they have default values that are indicated. This second part of the table presents and describes all the input information and the results that are associated with each web service implementation.

c. Status Codes

This part of the table presents and describes the list of standard response codes that can be produced by each web service, according to the API. The codes help identify the cause of the problem when the service is not working properly. The term HTTP status code is actually the common term for the HTTP status line that includes both the HTTP status code and the HTTP reason phrase. For example, the HTTP status line 500: Internal Server Error is made up of the HTTP status code of 500 and the HTTP reason phrase of Internal Server Error.

d. Implementation Information

The fourth part of the table provides technical details about the prototype implementation, such as the type of HTTP method that is used to execute the service, the programming languages and the open source libraries that were integrated into the service.

e. Examples

The last part of the table provides examples of the web service. The examples are based on the cURL command. They can be easily invoked from the command line and can be used by technical reviewers to test and evaluate the performance of each service.

5 Final Algorithm Documentation

In this chapter, first (section 5.1) we will give generic instructions how to use the different OpenTox algorithm web services that can be divided into descriptor calculation, model learning (classification and regression), feature selection, transformation, filtering, algorithms for estimating the applicability domain and miscellaneous algorithms that include for instances algorithms for structure generation or similarity and substructural search. Following that, we will present some models for REACH relevant endpoints (section 5.2), before we will list all algorithms that are available in the OpenTox framework and give a detailed description thereof (sections 5.3–5.6). This can be considered as a reference list for looking up or retrieving detailed information of certain algorithms.

5.1 Retrieving information and applying OpenTox algorithm web services

First, we will give a simple example to retrieve information that is needed to call an OpenTox algorithm web service. Second, we show the application of OpenTox algorithm web services.

5.1.1 Retrieving information of an algorithm

To get the information that is necessary to apply (call) an algorithm, e.g. to retrieve information about its parameters or default values, the corresponding algorithm URI can be queried via a curl GET call:

```
curl -X GET http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/algorithmID
```

Note that for this no parameter must be provided since solely information is queried. If, however, a protected resource URI is queried a token must be provided, resulting in:

```
curl -X GET http://opentox-dev.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/algorithmID -H "subjectid:someToken"
```

For example, if we want to call for instance gSpan, a frequent graph mining algorithm that delivers structural descriptors, we need the information which parameters are necessary for calling it. Via the following GET call we can query this information:

```
curl -X GET http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/gSpan
```

Amongst others, we retrieve the following piece of rdf/xml:

```
<ot:parameters>
  <ot:Parameter>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >minSup</dc:title>
    <dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Specifies the minimum support for mining (absolute) (Default: 20).</dc:description>
    <ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >optional</ot:paramScope>
    <ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >20</ot:paramValue>
  </ot:Parameter>
</ot:parameters>
```

It indicates that the parameter “minSup” (the most important parameter of gSpan) that **specifies the minimum support for mining has a default value of 20 and it is an optional parameter.** The complete output can be found in Appendix A.1.

5.1.2 Applying an algorithm

In the following, we will show how an algorithm can be called, or to put it differently from a technical point of view, how to create a new resource containing a learned model or a new dataset. For simplicity we assume that the user has chosen to take <http://apps.ideaconsult.net:8080/ambit2/dataset> as the dataset server, but also other servers like <https://ambit.uni-plovdiv.bg:8443/ambit2/dataset/>, a protected version for which a token is required that allows the user only to query data if they have privileges, or the dataset server in Freiburg <http://opentox.informatik.uni-freiburg.de/dataset> can be used. Further we assume that algorithms are used from TUM <http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm>, but also algorithms from AMBIT <http://apps.ideaconsult.net:8080/ambit2/algorithm>, NTUA <http://opentox.ntua.gr:8080/algorithm> or ALU <http://opentox.informatik.uni-freiburg.de/algorithm> can be used. Unless a dataset with descriptors is given, the user has to calculate descriptors that are essential for modelling. Furthermore, feature selection or algorithms for estimating the domain of applicability of a model can be applied. To this end OpenTox algorithms can be used that are presented in section 5.3–5.8. In general a cURL call looks like this:

```
curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/someDataset" -d
"dataset_server=http://apps.ideaconsult.net:8080/ambit2/dataset/" -d
"algorithmSpecificParameter1=SomeValue1" -d "algorithmSpecificParameter2= SomeValue2"
http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/algorithmID
```

Note that the signature to call all these different kinds of algorithms is the same. This underlines the generic nature of the OpenTox API and its framework, one of its strengths that enhances usability and extensibility. In the call above, POST indicated that an algorithm (indicated with algorithmID) is run to which parameters are posted and a new resource is created. With “-d” form parameters are indicated, with “-H” header parameters. On the left side of the equality sign the algorithm specific parameter name must be provided, whereas on the right side its value that can be set by the user is provided. Parameters can be optional or mandatory, often default values are provided. Always, a mandatory parameter represents the dataset that must be set by the user. For modelling, the prediction feature represents an additional mandatory parameter that specifies the endpoint or target value to be predicted. Optional parameters are typically algorithm-specific parameters for which generally default values are given, e.g. kernels for Support Vector Machines or minimum support values for graph mining tools like Free Tree Miner or gSpan. For more information on each algorithm, see sections 5.3–5.8. Header parameters are for instances necessary when tokens for Authentication and Authorisation come into play. If we want to use protected resources, i.e. instead of

<http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/algorithmID>

we want to use

<http://opentox-dev.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/algorithmID>,

we have to provide a token via -H “subjectid:someToken”.

The output of each kind of algorithm in the OpenTox framework is a URI. However, the content of the URI is different. More precisely, for descriptor calculation, a dataset URI is created which contains the original dataset with which the service was called, plus the calculated descriptors. For feature selection a dataset URI is also returned but with fewer (the selected) features of the original dataset. If something went wrong during descriptor calculation an error message is provided. Feature selection/transformation or filtering algorithms

work similarly and are omitted here for brevity. Modelling algorithms return a model URI pointing to the learned model and its meta-information.

Examples of applications of OpenTox algorithms for modelling REACH-relevant endpoints or just applying an algorithm are shown in the following.

5.2 Models for REACH relevant endpoints

For modelling REACH-relevant endpoints OpenTox partners provide a number of different models. We will discuss several of them in the following section.

5.2.1 Example TUM models

In this section we discuss two TUM models for two REACH relevant endpoints. We will show how the models were generated and explain why we decided to choose certain descriptors and algorithms.

5.2.1.1 KNN model for Caco-2 Permeability:

We provide a model for Caco-2 Permeability that was learned with a k nearest neighbour (knn) algorithm. We set k to 1 in order to take only the nearest neighbour of a test molecule into account when it comes to prediction. We used the default descriptors of dataset R545 that consist of two charged partial surface area descriptors, the topological polar surface area and the number of hydrogen bond donors. In particular the surface area descriptors should influence the endpoint value of a molecule considerably. And since molecules with similar properties should have similar permeability values, we decided to choose 1NN for modelling. With the following call we generated the model:

```
curl -d 'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545' -d
'prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200' - 'KNN=1' http://opentox-
dev.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/kNNregression -H
'subjectid:AQIC5wM2LY4SfcyAIH5Xc2wavVohtDzQY6hXrg97EUppVjM.*AAJTSQACMDE.*'
```

The call resulted in the following model URI:

http://opentox-dev.informatik.tu-muenchen.de:8080/OpenTox-dev/model/TUMOpenToxModel_kNN_19

We evaluated this model in a 10 fold cross-validation to get an estimate of its predictivity. We obtained a mean absolute error of 0.52. Further statistics were generated at the URI: <http://opentox.informatik.uni-freiburg.de/validation/crossvalidation/273>.

5.2.1.2 Decision tree model for Micronucleus Data

We decided to choose the decision tree learner J48 in order to obtain an interpretable model: http://opentox-dev.informatik.tu-muenchen.de:8080/OpenTox-dev/model/TUMOpenToxModel_j48_3. Note that it is relatively straightforward to generate rules out of a decision tree. Hence, interpretations of predictions can be easily deduced.

5.3 Descriptor Calculation Algorithms

5.3.1 FreeTreeMiner

The FreeTreeMiner (FTM) software computes all acyclic substructures (in mathematical terms: free or unrooted trees) occurring at a given minimum frequency in a set of molecules. The substructures are computed by a depth-first search. The frequent substructures are returned as SMARTS strings together with their occurrences in the given set of structures.

General Information about the service	
Service description	The FreeTreeMiner (FTM) software computes all acyclic substructures occurring at a given minimum frequency in a set of molecules.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/FTM (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/FTM (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Type of descriptor (substructural/physico-chemical, expressiveness: paths, trees, subgraphs, suitability for similarity/distance calculations?, ...)	Substructural descriptors, acyclic substructures, results can be used in all fingerprint-based similarity and distance measures.
Partner responsible for the implementation	Technische Universität München
Contact within OT	kramer@in.tum.de

Request/Response Information	
Posted Parameters	<p>dataset_uri (mandatory) (e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p> <p>minSup (optional, default: 0.8) All free trees that exceed the minimum support (minSup) are calculated.</p> <p>hydrogen (optional, default: 0) Determines whether hydrogen atoms are taken into account (0 = no, 1 = yes).</p>
Response	A task URI is provided if the service tries to calculate features. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes	
200	Success - The request has succeeded and the requested features were generated. The URI of the dataset is returned within the response body.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.

404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error – The parameters you have posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, received an unsuccessful response. In those cases, it seems that some other server is down.
503	The service is not available for the time – Try again later!

Examples	
Example	<pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -d "minSup=0.9" http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/FTM</pre>

5.3.2 FMiner

Fminer is a novel method for efficiently mining relevant tree-shaped subgraph descriptors with minimum frequency and correlation constraints, each representing a set of fragments sharing a common core structure (backbone), thereby reducing feature set size and runtime. The approach is able to optimize structural inter-feature entropy as opposed to occurrences, which is characteristic for open or closed fragment mining. In the experiments, the proposed method reduces feature set sizes by >90% and >30% compared to complete tree mining and open tree mining, respectively. Evaluation using cross validation runs shows that their classification accuracy is similar to the complete set of trees but significantly better than that of open trees. Compared to open or closed fragment mining, a large part of the search space can be pruned due to an improved statistical constraint (dynamic upper bound adjustment), which is also confirmed in the experiments in lower runtimes compared to ordinary (static) upper bound pruning. Further analysis using large-scale datasets yields insight into important properties of the proposed descriptors, such as dataset coverage and class size represented by each descriptor. A final cross validation run confirms that the novel descriptors render large training sets feasible which previously might have been intractable for computational models.

General Information about the service	
Service description	The FMiner software efficiently mines relevant tree-shaped subgraph descriptors with minimum frequency and correlation constraints.
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Type of descriptor (substructural/physico-chemical, expressiveness: paths, trees, subgraphs, suitability for similarity/distance calculations?, ...)	Substructural descriptors, acyclic substructures, currently no wildcards used or other more advanced features of the SMARTS language, results can be used in all fingerprint-based similarity and distance measures.
Partner responsible for the implementation	IST
Contact within OT	

maunza@fdm.uni-freiburg.de

Request/Response Information	
Posted Parameters	
<p>dataset_uri (mandatory)</p> <p>(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p> <p>miniminFrequency (optional)</p> <p>miniminCorrelation (optional)</p>	
Response	
<p>A task URI is provided if the service tries to calculate features. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.</p>	

Status Codes	
200	Success – The request has succeeded and the requested features were generated. The URI of the dataset is returned within the response body.
400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling.
500	Internal Server Error – The parameters you posted are acceptable but some internal error occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information	
HTTP Method	
POST	
Libraries used	
OpenBabel (open source), GSL	

5.3.3 gSpan'

The gSpan' algorithm implements two optimizations of the widely known gSpan algorithm for mining molecular databases. Both optimizations apply to the enumeration of subgraph occurrences in a graph database, which is, also according to our profiling, the most expensive operation of gSpan. The first optimization reduces the number of subgraph isomorphisms that need to be accessed for proper support computation in considering the symmetries inherent in many chemical molecules, and the second speeds up subgraph isomorphism tests by making use of the non-uniform frequency distribution of atom and bond types.

General Information about the service

Service description The gSpan' software is able to compute all paths, trees and graphs that occur at a given minimum frequency in a set of molecules.
URI http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/gSpan (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/gSpan (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Type of descriptor (substructural/physico-chemical, expressiveness: paths, trees, subgraphs, suitability for similarity/distance calculations?, ...) Substructural descriptors, currently no wildcards used or other more advanced features of the SMARTS language, results can be used in all fingerprint-based similarity and distance measures. The user can restrict the search to acyclic and/or linear fragments and/or fragments with a maximum number of edges (bonds).
Partner responsible for the implementation Technische Universität München
Contact within OT kramer@in.tum.de

Request/Response Information
Posted Parameters dataset_uri (mandatory) (e.g. dataset_uri= ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.
minSup (optional, default: 20) All free trees that are equal to or exceed the minimum support (minSup) are calculated.
embeddingLists (optional, default: 0) Use embedding lists.
symmetries (optional, default: 0) Use symmetries.
fragmentsWithMaxEdges (optional, default: 0) Restrict search to fragments with maximum i edges.
NumMaxEdges (optional, default: 5) Number of maximum i edges. (Feature corresponds to fragmentsWithMaxEdges; it can only be set if fragmentsWithMaxEdges is set to 1).
linearFragments (optional, default: 0) Restrict search to linear fragments.
acyclicFragments (optional, default: 0) Restrict search to acyclic fragments.

<p>Response</p> <p>A task URI is provided if the service tries to calculate features. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.</p>

Status Codes	
200	Success – The request has succeeded and the requested features were generated. The URI of the dataset is returned within the response body.
400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error – The parameters you posted are acceptable but some internal error occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information
<p>HTTP Method</p> <p>POST</p>
<p>Programming Language</p> <p>The project was built in Java and runs as a standalone application.</p>
<p>Libraries used</p> <p>gSpan' software package</p>

Examples
<p>Example</p> <pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -d "minSup=20" http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/gSpan</pre>

5.3.4 MakeMNA

MakeMNA is a software product for generating MNA descriptors.

These descriptors are based on the molecular structure representation, which includes the hydrogens according to the valences and partial charges of other atoms and does not specify the types of bonds. MNA descriptors are generated as recursively defined sequence:

- zero-level MNA descriptor for each atom is the mark A of the atom itself;
- any next-level MNA descriptor for the atom is the sub-structure notation $A(D_1D_2..D_i...)$, where D_i is the previous-level MNA descriptor for i -th immediate neighbours of the atom A.

The mark of atom may include not only the atomic type but also any additional information about the atom. In particular, if the atom is not included into the ring, it is marked by “-”. The neighbour descriptors $D_1D_2..D_i...$

are arranged in unique manner, e.g., in lexicographic order. Iterative process of MNA descriptors generation can be continued covering first, second, etc. neighbourhoods of each atom.

General Information about the service	
Service description	The MakeMNA web service enables the user to calculate Multilevel Neighbourhoods of Atom (MNA) descriptors.
URI	http://195.178.207.160/OpenTox/algorithm/MakeMNA (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Type of descriptor (substructural/physico-chemical, expressiveness: paths, trees, subgraphs, suitability for similarity/distance calculations?, ...)	Substructural
Partner responsible for the implementation	Institute of Biomedical Chemistry of Russian Academy of Medical Sciences
Contact within OT	Dmitry.druzhilovsky@ibmc.msk.ru

Request/Response Information	
Posted Parameters	<p>dataset_uri (mandatory)</p> <p>(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p>
Response	Once the features for a dataset are generated successfully, datasets URI is returned to the client within the response and the status is set to 200; otherwise an explanatory message is provided.

Status Codes	
200	Success – The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://195.178.207.160/OpenTox/algorithm/
500	Internal Server Error – The parameters you posted are acceptable but some internal error occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.

503	The service is not available for the time being – Try again later!
-----	--

Implementation Information
HTTP Method POST
Programming Language The project was built in Delphi and runs as a standalone application.
Libraries used Embarcadero Delphi 2007 software package

Examples
Example <pre>curl -X POST -d dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/2765 -d dataset_service=http://apps.ideaconsult.net:8080/ambit2/dataset -H subjectid:token http://195.178.207.160/OpenTox/Algorithm/MakeMNA</pre>

5.3.5 MakeQNA

MakeQNA is a software product for generating QNA descriptors. Quantitative Neighbourhoods of Atoms (QNA) descriptors are based on quantities of ionization potential (IP) and electron affinity (EA) of each atom of the molecule. They are calculated as follows:

- $P_i = B_i - \frac{1}{2} \sum_k (\exp(-\frac{1}{2}C))_{ik} B_k - \frac{1}{2}$,
- $Q_i = B_i - \frac{1}{2} \sum_k (\exp(-\frac{1}{2}C))_{ik} B_k - \frac{1}{2} A_k$,
- $A_i = \frac{1}{2}(I_{P_i} + E_{A_i})$, $B_i = I_{P_i} - E_{A_i}$,

Where I_{P_i} is the ionization potential (the energy required to remove the outermost electron from a neutral gaseous atom), and E_{A_i} is the electron affinity (the energy released when an electron is added to a neutral gaseous atom of that element) of atom i .

General Information about the service
Service description The MakeQNA web service enables the user to calculate Quantitative Neighbourhoods of Atoms (QNA) descriptors.
URI http://195.178.207.160/OpenTox/algorithm/MakeQNA (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Type of descriptor (substructural/physico-chemical, expressiveness: paths, trees, subgraphs, suitability for similarity/distance calculations?, ...) Numerical reflecting the interatomic interaction for each atom in a molecule.
Partner responsible for the implementation

Institute of Biomedical Chemistry of Russian Academy of Medical Sciences

Contact within OT

Dmitry.druzhilovsky@ibmc.msk.ru

Request/Response Information

Posted Parameters

dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.

Response

Once the features for a dataset are generated successfully, datasets URI is returned to the client within the response and the status is set to 200; otherwise an explanatory message is provided.

Status Codes

200	Success – The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://195.178.207.160/OpenTox/algorithm/
500	Internal Server Error – The parameters you posted are acceptable but some internal error occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information

HTTP Method

POST

Programming Language

The project was built in Delphi and runs as a standalone application.

Libraries used

Embarcadero Delphi 2007 software package

Examples

Example

```
curl -X POST -d dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/2765 -d
dataset_service=http://apps.ideaconsult.net:8080/ambit2/dataset -H subjectid:token
http://195.178.207.160/OpenTox/Algorithm/MakeQNA
```

5.3.6 JOELib2

JOELib2 is a platform-independent open source computational chemistry package written in Java. JOELib2 consists of an algorithm library that was designed for prototyping, data mining and graph mining of chemical compounds. JOELib2 is the Java successor of the JOELib library from OpenEye.

General Information about the service	
Service description	The JOELIB2 web service enables the user to calculate physicochemical, geometrical descriptors, functional groups, atom properties and fingerprints.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/JOELIB2 (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/JOELIB2 (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Type of descriptor (substructural/physico-chemical, expressiveness: paths, trees, subgraphs, suitability for similarity/distance calculations?, ...)	Physicochemical, geometrical descriptors, functional groups, atom properties, fingerprints, transformations (see Tutorial pages 24-35 www.ra.cs.uni-tuebingen.de/software/joelib/tutorial/JOELibTutorial.pdf)
Partner responsible for the implementation	Technische Universität München
Contact within OT	kramer@in.tum.de

Request/Response Information	
Posted Parameters	
dataset_uri (mandatory)	(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.
ALL (optional, default: true)	Calculate all available descriptors.
AcidicGroups (optional, default: false)	Number of acidic groups.
AliphaticOHGroups (optional, default: false)	Number of aliphatic hydroxy groups.
AromaticOHGroups (optional, default: false)	Number of aromatic hydroxy groups.
AromaticBonds (optional, default: false)	

Number of aromatic bonds.

BasicGroups (optional, default: false)

Number of basic groups.

FractionRotatableBonds (optional, default: false)

Fraction of rotatable bonds.

GeometricalDiameter (optional, default: false)

Calculates the geometrical diameter.

GeometricalRadius (optional, default: false)

Calculates the geometrical radius.

GeometricalShapeCoefficient (optional, default: false)

Calculates the geometrical shape coefficient.

GlobalTopologicalChargeIndex (optional, default: false)

Calculates the Topological Charge Index.

GraphShapeCoefficient (optional, default: false)

Calculates the graph shape coefficient.

HeavyBonds (optional, default: false)

Number of heavy bonds.

HeteroCycles (optional, default: false)

Number of hetero cycles.

HydrophobicGroups (optional, default: false)

Number of hydrophobic groups.

KierShape1 (optional, default: false)

Calculates the Kier Shape for paths with length one.

KierShape2 (optional, default: false)

Calculates the Kier Shape for paths with length two.

KierShape3 (optional, default: false)

Calculates the Kier Shape for paths with length three.

LogP (optional, default: false)

Calculates the Octanol/Water partition coefficient (logP) or hydrophobicity.

MolarRefractivity (optional, default: false)

Calculates the molar refractivity (MR).

MolecularWeight (optional, default: false)

Calculates the molecular weight.

NumberOfAtoms (optional, default: false)

Number of atoms.

NumberOfBr (optional, default: false)

Number of bromium atoms.

NumberOfBonds (optional, default: false)

Number of bonds.

NumberOfC (optional, default: false)

Number of carbon atoms.

NumberOfCl (optional, default: false)

Number of chlorine atoms.

NumberOfHal (optional, default: false)

Number of halogen atoms.

NumberOfI (optional, default: false)

Number of iodine atoms.

NumberOfF (optional, default: false)

Number of fluorine atoms.

NumberOfN (optional, default: false)

Number of nitrogen atoms.

NumberOfO (optional, default: false)

Number of oxygen atoms.

NumberOfP (optional, default: false)

Number of phosphorus atoms.

NumberOfS (optional, default: false)

Number of sulfur atoms.

NO2Groups (optional, default: false)

Number of NO₂ groups.

OSOGroups (optional, default: false)

Number of OSO groups.

RotatableBonds (optional, default: false)

Number of rotatable bonds.

SOGroups (optional, default: false)

Number of SO₂ groups.

SO2Groups (optional, default: false)

Number of SO groups.

TopologicalDiameter (optional, default: false)

Calculates the topological diameter.

TopologicalRadius (optional, default: false)

Calculates the topological radius.

SSKey3DS (optional, default: false)

Pharmacophore fingerprint.

Response

A task URI is provided if the service tries to calculate features. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes

200	Success – The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information

HTTP Method

POST

Programming Language

The project was built in Java and runs as a standalone application.

Libraries used

The service uses the JOELIB2 software package.

Examples

Example 1

```
curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -d "ALL=false"
-d "NumberOfF=true" -d "KierShape3=true" -d "FractionRotatableBonds=true"
http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/JOELIB2
```

5.3.7 OpenBabel

Open Babel is a chemical toolbox designed to speak the many languages of chemical data. It is an open, collaborative project allowing anyone to search, convert, analyse, or store data from molecular modelling, chemistry, solid-state materials, biochemistry, or related areas. OpenBabel is an open source computational chemistry package written in C++. The software is available for the Linux, Windows and MAC operating

system. The OpenBabel implementation has no dependencies on other software packages. For further information, we refer to the OpenBabel website openbabel.org.

General Information about the service	
Service description	The OpenBabel web service enables the user to calculate a variety of physicochemical and other descriptors.
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Type of descriptor (substructural/physico-chemical, expressiveness: paths, trees, subgraphs, suitability for similarity/distance calculations?, ...)	Substructural, physicochemical, topological, etc.
Partner responsible for the implementation	IST

Request/Response Information	
Posted Parameters	<p>dataset_uri (mandatory)</p> <p>(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p>
Response	A task URI is provided if the service tries to calculate features. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes	
200	Success – The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling.
500	Internal Server Error – The parameters you have posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information

HTTP Method	POST
Libraries used	The service uses the OpenBabel software package.

5.3.8 3D structure generation, based on MOPAC (Molecular Orbital PACKage)

General Information about the service	
Service description	Given a dataset with 2D structures, generates optimized 3D structure, and replaces the original ones. Given a dataset with 3D structures, optimizes the structures and replaces ye original ones.
URI	http://apps.ideaconsult.net:8080/ambit2/algorithm/ambit2.mopac.MopacShell
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Structure generation/optimization
Partner responsible for the implementation	Ideaconsult Ltd.
Contact within OT	Nina Jeliaskova <jeliaskova.nina@gmail.com>

Request/Response Information	
Posted Parameters	<p>dataset_uri (mandatory)</p> <p>The data set URI is a mandatory parameter that has to be specified by the client. When assessing a training set of a model, this is the training set of the model.</p> <p>param (optional, algorithm specific parameters, as defined by MOPAC command options)</p>
Response	A task is created and returned to the client which upon successful completion will be pointing to the URI of the dataset with optimized structures

Status Codes	
202	Accepted - The request has succeeded and a task is returned to the client.
400	Bad Request - Invalid parametrization;
404	The resource was not found - check your spelling.
500	Internal Server Error - The parameters you posted are acceptable but some internal error has

occurred. This might occur if MOPAC could not be found.

Implementation Information
HTTP Method POST
Programming Language The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://apps.ideaconsult.net:8080/ambit2 runs within an Apache Tomcat container.
Libraries used Smi23d and mengine are used to generate initial (non-optimized) 3D structure. Optimization is performed by MOPAC. The full list of dependencies is available via maven pom.xml files of AMBIT project.

Examples
Example <pre>curl -X POST -d dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545 -d -H subjectid:YOUR-TOKEN http://apps.ideaconsult.net:8080/ambit2/algorithm/ambit2.mopac.MopacShell</pre>

5.3.9 Semiempirical quantum chemistry descriptors, based on MOPAC (Molecular Orbital PACKage)

General Information about the service
Service description Given a dataset with 2D structures, calculates electronic descriptors NO. OF FILLED LEVELS, TOTAL ENERGY, FINAL HEAT OF FORMATION, IONIZATION POTENTIAL, ELECTRONIC ENERGY, CORE-CORE REPULSION, MOLECULAR WEIGHT, EHOMO, ELUMO, by running MOPAC. Does not attempt to optimize the structure(s). Fails, if no 3D structure is available.
URI http://apps.ideaconsult.net:8080/ambit2/algorithm/ambit2.mopac.DescriptorMopacShell
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Descriptor calculation, electronic descriptors
Partner responsible for the implementation Ideaconsult Ltd.
Contact within OT Nina Jeliaskova <jeliaskova.nina@gmail.com>

Request/Response Information

<p>Posted Parameters</p> <p>dataset_uri (mandatory)</p> <p>The data set URI is a mandatory parameter that has to be specified by the client. When assessing a training set of a model, this is the training set of the model.</p> <p>param (optional, algorithm specific parameters, as defined by MOPAC command options)</p>
<p>Response</p> <p>A task, which, upon successful completion, will be pointing to the URI of the dataset with calculated descriptors, is created and returned to the client.</p>

Status Codes	
202	Accepted - The request has succeeded and a task is returned to the client.
400	Bad Request - Invalid parametrization;
404	The resource was not found - check your spelling.
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred. This might occur if MOPAC could not be found.

Implementation Information
<p>HTTP Method</p> <p>POST</p>
<p>Programming Language</p> <p>The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://apps.ideaconsult.net:8080/ambit2 runs within an Apache Tomcat container.</p>
<p>Libraries used</p> <p>Descriptor calculation is performed by MOPAC. The full list of dependencies is available via maven pom.xml files of AMBIT project.</p>

Examples
<p>Example</p> <pre>curl -X POST -d dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545 -d -H subjectid:YOUR-TOKEN http://apps.ideaconsult.net:8080/ambit2/algorithm/ambit2.mopac.DescriptorMopacShell</pre>

5.3.10 AMBIT

AMBIT descriptor calculation web service offers a list of descriptor calculation algorithms, implemented by several packages, including the CDK, AMBIT, DRAGON, MOPAC. Adding a new descriptor calculation algorithm requires only creating a wrapper, exposing the native implementation via the CDK IMolecularDescriptor interface.

AMBIT RESTful web services: an implementation of the OpenTox application programming interface, Journal of Cheminformatics. 2011, 3:18doi:10.1186/1758-2946-3-18. <http://www.jcheminf.com/content/3/1/18>

General Information about the service	
Service description	<p>Given a dataset with 2D or 3D structures, calculates different types of descriptors</p> <p>48 descriptors, list available under http://apps.ideaconsult.net:8080/ambit2/algorithm?type=Rules</p>
URI	<p>Example:</p> <p>http://apps.ideaconsult.net:8080/ambit2/algorithm/org.openscience.cdk.qsar.descriptors.molecular.XLogPDescriptor</p>
OpenTox API Reference	<p>www.opentox.org/dev/apis/api-1.2/Algorithm</p>
Algorithm type	<p>Descriptor calculation</p>
Partner responsible for the implementation	<p>Ideaconsult Ltd.</p>
Contact within OT	<p>Nina Jeliaskova <jeliaskova.nina@gmail.com></p>

Request/Response Information	
Posted Parameters	<p>dataset_uri (mandatory)</p> <p>The data set URI is a mandatory parameter that has to be specified by the client. When assessing a training set of a model, this is the training set of the model.</p> <p>param (optional, algorithm specific parameters)</p>
Response	<p>A task, which, upon successful completion, will be pointing to the URI of the dataset with calculated descriptors, is created and returned to the client.</p>

Status Codes	
202	Accepted - The request has succeeded and a task is returned to the client.
400	Bad Request - Invalid parametrization;
404	The resource was not found - check your spelling.
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.

Implementation Information	
HTTP Method	

POST
Programming Language The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://apps.ideaconsult.net:8080/ambit2 runs within an Apache Tomcat container.
Libraries used Descriptor calculation is performed by the CDK library (v.3.1.18) , AMBIT 2 (2.4.1), Dragon 6. The full list of dependencies is available via maven pom.xml files of AMBIT project.

Examples
Example <pre>curl -X POST -d dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545 -d -H subjectid:YOUR-TOKEN http://apps.ideaconsult.net:8080/ambit2/algorithm/org.openscience.cdk.qsar.descriptors.molecular.XLogPDescriptor</pre>

5.3.11 The Chemistry Development Kit (CDK)

The Chemistry Development Kit (CDK) is a Java library for structural chemo- and bioinformatics. It is now developed by more than 50 developers all over the world and used in more than 10 different academic as well as industrial projects worldwide. A number of descriptor implementations are available.

General Information about the service
Service description The CDK web service enables the user to calculate a variety of physicochemical and other descriptors.
URI http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/CDKPhysChem (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/CDKPhysChem (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Type of descriptor (substructural/physico-chemical, expressiveness: paths, trees, subgraphs, suitability for similarity/distance calculations?, ...) Substructural, physicochemical, topological, etc.
Partner responsible for the implementation Technische Universität München
Contact within OT kramer@in.tum.de

Request/Response Information

Posted Parameters
dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.

ALL (optional, default: true)

Calculate all available descriptors.

ELECTRONIC (optional, default: false)

Calculate all available electronic descriptors.

GEOMETRICAL (optional, default: false)

Calculate all available geometrical descriptors.

CONSTITUTIONAL (optional, default: false)

Calculate all available constitutional descriptors.

HYBRID (optional, default: false)

Calculate all available hybrid descriptors.

TOPOLOGICAL (optional, default: false)

Calculate all available topological descriptors.

ALOGPDescriptor (optional, default: false)

Calculates atom additive logP and molar refractivity values as described by Ghose and Crippen.

APolDescriptor (optional, default: false)

Descriptor that calculates the sum of the atomic polarizabilities (including implicit hydrogens).

AminoAcidCountDescriptor (optional, default: false)

Returns the number of amino acids found in the system.

AromaticAtomsCountDescriptor (optional, default: false)

Descriptor based on the number of aromatic atoms of a molecule.

AromaticBondsCountDescriptor (optional, default: false)

Descriptor based on the number of aromatic bonds of a molecule.

AtomCountDescriptor (optional, default: false)

Descriptor based on the number of atoms of a certain element type.

AutocorrelationDescriptorCharge (optional, default: false)

The Moreau–Broto autocorrelation descriptors using partial charges.

AutocorrelationDescriptorMass (optional, default: false)

The Moreau–Broto autocorrelation descriptors using atomic weight.

AutocorrelationDescriptorPolarizability (optional, default: false)

The Moreau–Broto autocorrelation descriptors using polarizability.

BCUTDescriptor (optional, default: false)

Eigenvalue-based descriptor noted for its utility in chemical diversity described by Pearlman et al.

BPolDescriptor (optional, default: false)

Descriptor that calculates the sum of the absolute value of the difference between atomic polarizabilities of all bonded atoms in the molecule (including implicit hydrogens).

BondCountDescriptor (optional, default: false)

Descriptor based on the number of bonds of a certain bond order.

CPSADescriptor (optional, default: false)

A variety of descriptors combining surface area and partial charge information.

CarbonTypesDescriptor (optional, default: false)

Characterizes the carbon connectivity in terms of hybridization.

ChiChainDescriptor (optional, default: false)

Evaluates the Kier & Hall Chi chain indices of orders 3,4,5 and 6.

ChiClusterDescriptor (optional, default: false)

Evaluates the Kier & Hall Chi cluster indices of orders 3,4,5,6 and 7.

ChiPathClusterDescriptor (optional, default: false)

Evaluates the Kier & Hall Chi path cluster indices of orders 4,5 and 6.

ChiPathDescriptor (optional, default: false)

Evaluates the Kier & Hall Chi path indices of orders 0,1,2,3,4,5,6 and 7.

EccentricConnectivityIndexDescriptor (optional, default: false)

A topological descriptor combining distance and adjacency information.

FragmentComplexityDescriptor (optional, default: false)

Class that returns the complexity of a system. The complexity is defined as @cdk.cite{Nilakantan06}.

GravitationalIndexDescriptor (optional, default: false)

Descriptor characterizing the mass distribution of the molecule.

HBondAcceptorCountDescriptor (optional, default: false)

Descriptor that calculates the number of hydrogen bond acceptors.

HBondDonorCountDescriptor (optional, default: false)

Descriptor that calculates the number of hydrogen bond donors.

KappaShapeIndicesDescriptor (optional, default: false)

Descriptor that calculates Kier and Hall kappa molecular shape indices.

KierHallSmartsDescriptor (optional, default: false)

Counts the number of occurrences of the E-state fragments.

LargestChainDescriptor (optional, default: false)

Returns the number of atoms in the largest chain.

LargestPiSystemDescriptor (optional, default: false)

Returns the number of atoms in the largest pi chain.

LengthOverBreadthDescriptor (optional, default: false)

Calculates the ratio of length to breadth.

LongestAliphaticChainDescriptor (optional, default: false)

Returns the number of atoms in the longest aliphatic chain.

MDEDescriptor (optional, default: false)

Evaluates molecular distance edge descriptors for C, N and O.

MomentOfInertiaDescriptor (optional, default: false)

Descriptor that calculates the principal moments of inertia and ratios of the principal moments. Also calculates the radius of gyration.

PetitjeanNumberDescriptor (optional, default: false)

Descriptor that calculates the Petitjean Number of a molecule.

PetitjeanShapeIndexDescriptor (optional, default: false)

The topological and geometric shape indices described by Petitjean and Bath et al. respectively. Both measure the anisotropy in a molecule.

RotatableBondsCountDescriptor (optional, default: false)

Descriptor that calculates the number of non-rotatable bonds on a molecule.

RuleOfFiveDescriptor (optional, default: false)

This Class contains a method that returns the number of failures of the Lipinski's Rule Of Five.

TPSADescriptor (optional, default: false)

Calculation of topological polar surface area based on fragment contributions.

VAdjMaDescriptor (optional, default: false)

Descriptor that calculates the vertex adjacency information of a molecule.

WHIMDescriptor (optional, default: false)

Holistic descriptors described by Todeschini et al.

WeightDescriptor (optional, default: false)

Descriptor based on the weight of atoms of a certain element type. If no element is specified, the returned value is the Molecular Weight.

WeightedPathDescriptor (optional, default: false)

The weighted path (molecular ID) descriptors described by Randic. They characterize molecular branching.

WienerNumbersDescriptor (optional, default: false)

This class calculates Wiener path number and Wiener polarity number.

XLogPDescriptor (optional, default: false)

Prediction of logP based on the atom-type method called XLogP.

ZagrebIndexDescriptor (optional, default: false)

The sum of the squared atom degrees of all heavy atoms.

Response

A task URI is provided if the service tries to calculate features. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes

200	Success – The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error – The parameters you have posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information

HTTP Method

POST

Programming Language

The project was built in Java and runs as a standalone application.

Libraries used

The service uses the CDK software package.

Examples

Example 1

```
curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -d
"HBondAcceptorCountDescriptor=true" -d "ALL=false" -d "ELECTRONIC=true"
http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/CDKPhysChem
```

5.4 Classification and Regression Algorithms

5.4.1 Gaussian Processes for Regression

GPR (Gaussian Processes for Regression) is a method of supervised learning. A Gaussian process is a generalization of the Gaussian probability distribution. Whereas a probability distribution describes random variables which are scalars or vectors (for multivariate distributions), a stochastic process governs the properties of functions. Just as a Gaussian distribution is fully specified by its mean and covariance matrix, a Gaussian process is specified by a mean and covariance function. Here, the mean is a function of x (which we

will often take to be the zero function), and the covariance is a function $C(x, x')$ that expresses the expected covariance between the values of the function y at the points x and x' . The function $y(x)$ in any one data modelling problem is assumed to be a single sample from this Gaussian distribution.

General Information about the service	
Service description	The Gaussian Process web service enables the user to build regression models for a specific data set.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/GaussP (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/GaussP (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Regression
Partner responsible for the implementation	Technische Universität München
Contact within OT	kramer@in.tum.de

Request/Response Information	
Posted Parameters	
dataset_uri (mandatory)	(e.g. dataset_uri= ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.
kernel (optional, default: PolyKernel)	The kernel to use options are PolyKernel, Puk and RBFKernel.
gamma (optional, default: 1.0)	Parameter for the rbf kernel only.
omega (optional, default: 1.0)	Parameter for the puk kernel only.
sigma (optional, default: 1.0)	Parameter for the puk kernel only.
exponent (optional, default: 1.0)	Parameter for the polynomial kernel only.
noise (optional, default: 1.0)	Whether to use unsmoothed predictions.

<p>Response</p> <p>A task URI is provided if the service tries to calculate a regression model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.</p>

Status Codes	
200	Success – The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect – the result can be found elsewhere.
400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, received an unsuccessful response. In those cases, it seems that some other server is down.
503	The service is not available for the time – Try again later!

Implementation Information
<p>HTTP Method</p> <p>POST</p>
<p>Programming Language</p> <p>The project was built in Java and runs as a standalone application.</p>
<p>Libraries used</p> <p>The service uses the WEKA software package.</p>

Examples
<p>Example</p> <pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/598242" -d "prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200" http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/GaussP</pre>

5.4.2 Lazar

Lazar is a k-nearest-neighbour approach to predict chemical endpoints from a training set based on structural fragments. It uses a SMILES file and precomputed fragments with occurrences as well as target class information for each compound as training input. It also features regression, in which case the target activities consist of continuous values. Lazar uses activity-specific similarity (i.e. each fragment contributes with its significance for the target activity) that is the basis for predictions and confidence index for every single prediction. For classification, a weighted nearest neighbour voting is the standard prediction, whereas for

regression a kernel model based on activity-specific similarity is used by default. A kernel model is also available for classification, as well as a multilinear model for regression.

General Information about the service	
Service description	The lazar web services enable the user to build regression/classification models for a specific data set.
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Regression, Classification
Partner responsible for the implementation	IST
Contact within OT	helma@in-silico.de , maunza@fdm.uni-freiburg.de

Request/Response Information	
Posted Parameters	<p>dataset_uri (mandatory)</p> <p>(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p>
Response	A task URI is provided if the service tries to calculate a model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes	
200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling.
500	Internal Server Error - The parameters you posted have are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being - Try again later!

Implementation Information
HTTP Method POST

5.4.3 KNN

The k -nearest neighbours algorithm (knn) is a method for classifying objects based on closest training examples in the feature space. It is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is delayed until classification. A majority vote of an object's neighbours is used for classification, with the object being assigned to the class most common amongst its k (positive integer, typically small) nearest neighbours. If k is set to 1, then the object is simply assigned to the class of its nearest neighbour. The knn algorithm can also be applied for regression in the same way by simply assigning the property value for the object to be the average of the values of its k nearest neighbours. It can be useful to weight the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones. No explicit training step is required since training consists of just storing training instance feature vectors and corresponding class labels. In order to identify neighbours, the objects are represented by position vectors in a multidimensional feature space. It is usual to use the Euclidean distance, but also further distance measures, such as the Manhattan distance could be used instead. In the classification/testing phase, the test sample is represented as a vector in the feature space. Distances from this vector to all stored vectors are computed and the k closest samples are selected to determine the class/real-value of the test instance.

The k -nearest neighbour algorithm is sensitive to the local structure of the data. The best choice of k depends upon the data; generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good k can be selected by various heuristic techniques like cross-validation. The accuracy of the kNN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance.

General Information about the service
Service description The kNNregression and kNNclassification web services enable the user to build regression/classification models for a specific data set using the lazy nearest neighbour approach.
URI http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/kNNregression (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/kNNregression (AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/kNNclassification (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/kNNclassification (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Regression, Classification
Partner responsible for the implementation Technische Universität München

Contact within OT

kramer@in.tum.de

Request/Response Information

Posted Parameters

dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.

KNN (optional, default: 1)

set the k parameter - number of neighbours considered.

distanceWeighting (optional, default: 0)

0 for no distance weighting, I for 1/distance or F for 1-distance.

meanSquared (optional, default: 0)

Whether the mean squared error is used rather than mean absolute error when doing cross-validation for regression problems.

windowSize (optional, default: 0)

Gets the maximum number of instances allowed in the training pool. The addition of new instances above this value will result in old instances being removed. A value of 0 signifies no limit to the number of training instances.

Response

A task URI is provided if the service tries to calculate a model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes

200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error - The parameters you posted have are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being - Try again later!

Implementation Information

HTTP Method	POST
Programming Language	The project was built in Java and runs as a standalone application.
Libraries used	The service uses the WEKA software package.

Examples
Example
<pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/598242" -d "prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200" http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/kNNregression</pre>

5.4.4 J48

J48 implements Quinlan's C4.5 algorithm for generating a pruned or unpruned C4.5 decision tree. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by J48 can be used for classification. J48 builds decision trees from a set of labelled training data using the concept of information entropy. It uses the fact that each attribute of the data can be used to make a decision by splitting the data into smaller subsets. J48 examines the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. To make the decision, the attribute with the highest normalized information gain is used. Then the algorithm recurs on the smaller subsets. The splitting procedure stops if all instances in a subset belong to the same class. Then a leaf node is created in the decision tree telling to choose that class. But it can also happen that none of the features give any information gain. In this case J48 creates a decision node higher up in the tree using the expected value of the class.

J48 can handle both continuous and discrete attributes and training data with missing attribute values. Further it provides an option for pruning trees after creation.

General Information about the service
Service description
The J48 web service enables the user to build classification models for a specific data set with the C4.5 algorithm.
URI
http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/J48 (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/J48 (AA)
OpenTox API Reference
www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type
Classification
Partner responsible for the implementation
Technische Universität München

Contact within OT

kramer@in.tum.de

Request/Response Information
Posted Parameters
dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.

binarySplits (optional, default: 0)

Use binary splits on nominal attributes when building trees.

ConfidenceFactor (optional, default: 0.25)

The confidence factor used for pruning (smaller values incur more pruning).

minNumObj (optional, default: 2)

Minimum number of instances per leaf.

numFolds (optional, default: 3)

Determines the amount of data used for reduced-error pruning. One fold is used for pruning, the rest for growing the tree.

reducedErrorPruning (optional, default: 0)

Whether reduced-error pruning is used instead of C.4.5 pruning.

seed (optional, default: 1)

The seed used for randomizing the data when reduced-error pruning is used.

subtreeRaising (optional, default: 1)

Whether to consider the subtree raising operation when pruning.

unpruned (optional, default: 0)

Whether pruning is performed.

useLaplace (optional, default: 0)

Whether counts at leaves are smoothed based on Laplace.

Response

A task URI is provided if the service tries to calculate a classification model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes

200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.

404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error – The parameters you have posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information	
HTTP Method	POST
Programming Language	The project was built in Java and runs as a standalone application.
Libraries used	The service uses the WEKA software package.

Examples	
Example	<pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/585758" -d "prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/111148" http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/J48</pre>

5.4.5 M5P

M5P is a reconstruction of Quinlan's M5 algorithm for inducing trees of regression models. M5P combines a conventional decision tree with the possibility of linear regression functions at the nodes. First, a decision-tree induction algorithm is used to build a tree, but instead of maximizing the information gain at each inner node, a splitting criterion is used that minimizes the intra-subset variation in the class values down each branch. The splitting procedure in M5P stops if the class values of all instances that reach a node vary very slightly, or only a few instances remain. Second, the tree is pruned back from each leaf. When pruning an inner node is turned into a leaf with a regression plane. Third, to avoid sharp discontinuities between the subtrees a smoothing procedure is applied that combines the leaf model prediction with each node along the path back to the root, smoothing it at each of these nodes by combining it with the value predicted by the linear model for that node. M5P generates models that are compact and relatively comprehensible.

General Information about the service	
Service description	The M5P web service enables the user to build regression models for a specific data set with the M5 algorithm by R. Quinlan and Yong Wang.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/M5P (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/M5P (AA)

OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Regression
Partner responsible for the implementation Technische Universität München
Contact within OT kramer@in.tum.de

Request/Response Information
Posted Parameters <p>dataset_uri (mandatory) (e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p> <p>buildRegressionTree (optional, default: 0) Whether to generate a regression tree/rule instead of a model tree/rule.</p> <p>minNumInstances (optional, default: 4.0) The minimum number of instances to allow at a leaf node. Must be an integer greater than 0.</p> <p>useUnsmoothed (optional, default: 0) Whether to use unsmoothed predictions.</p> <p>unpruned (optional, default: 0) Whether unpruned tree/rules are to be generated.</p>
Response A task URI is provided if the service tries to calculate a regression model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes	
200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error - The parameters you have posted are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it

	received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information	
HTTP Method	POST
Programming Language	The project was built in Java and runs as a standalone application.
Libraries used	The service uses the WEKA software package.

Examples	
Example	<pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/598242" -d "prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200" http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/M5P</pre>

5.4.6 Fuzzy-means

Fuzzy-means is a training method for Radial Basis Function (RBF) neural networks and is based on the fuzzy partition of the input space, which is produced by defining a number of triangular fuzzy sets in the domain of each input variable. The centers of these fuzzy sets form a multidimensional grid on the input space. A rigorous selection algorithm chooses the most appropriate vertices on the grid, which are then used as the hidden node centers in the resulting RBF network model. The so called “fuzzy-means” training method does not need the number of centers to be fixed before the execution of the method. Due to the fact that it is a one-pass algorithm, it is extremely fast, even in the case of a large database of input-output training data. The method was originally developed for solving nonlinear regression problems. A variant of the method for solving classification problems has also been developed.

General Information about the service	
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Regression
Partner responsible for the implementation	NTUA
Contact within OT	hsarimv@central.ntua.gr

Request/Response Information	
Posted Parameters	

<p>dataset_uri (mandatory)</p> <p>(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545 which should be available in RDF format.</p>
<p>Response</p> <p>A task URI is provided if the service tries to calculate a regression model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.</p>

Status Codes	
200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling.
500	Internal Server Error - The parameters you have posted are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being - Try again later!

Implementation Information
<p>HTTP Method</p> <p>POST</p>
<p>Libraries used</p> <p>Matlab</p>

5.4.7 MakeSCR

Self-consistent regression (SCR)

Delphi implementation of a self-consistent regression algorithm. Using self-consistent regression one can obtain the best QSAR/QSPR model for the training set with a large number of descriptors. SCR is based on least-squares regularized method. The main features of SCR are the following:

- variable selection
- model building
- parameters of model calculation (R2, Q2, SD, Fisher)
- validation by LOOCV
- y-scrambling

General Information about the service
<p>Service description</p>

<p>The MakeSCR web service enables the user to create regression equation based on self-consistent regression algorithm.</p>
<p>URI</p> <p>http://195.178.207.160/OpenTox/algorithm/MakeSCR</p>
<p>OpenTox API Reference</p> <p>www.opentox.org/dev/apis/api-1.2/Algorithm</p>
<p>Algorithm type</p> <p>Regression</p>
<p>Partner responsible for the implementation</p> <p>Institute of Biomedical Chemistry of Russian Academy of Medical Sciences</p>
<p>Contact within OT</p> <p>alexey.zakharov@ibmc.msk.ru</p>

Request/Response Information

Posted Parameters

dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.

prediction_feature (mandatory)

A feature among the ones in the dataset submitted to the service as dataset_uri. Necessarily must be of type 'Numeric'.

Response

A task URI is provided if the service tries to calculate a regression model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes

200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Invalid parameterization; common mistakes are that the dataset URI was misspelled of the feature URI was not included in the dataset. Check under /dataset/id/feature for a list of features available. A 400 will also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled). Invalid parameterization can also occur if optional parameters are set to non-meaningful values (e.g. a negative value of epsilon).
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://195.178.207.160/OpenTox/algorithm/
500	Internal Server Error - The parameters you have posted are acceptable but some internal error has occurred.

502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information	
HTTP Method	POST
Programming Language	The project was built in Delphi and PHP code.
Libraries used	The service uses the SQL software.

Examples	
Example	<pre>curl -iv -X POST -d dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/601261 -d prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/534498 -H subjectid:token http://195.178.207.160/OpenTox/algorithm/MakeSCR</pre>

5.4.8 ToxTree

ToxTree is a full-featured and flexible user-friendly open source application, which is able to estimate toxic hazard by applying a decision tree approach. Currently it includes the following modules:

1. Cramer rules [CRA78]
2. Verhaar scheme for predicting toxicity mode of actions [VER92]
3. A decision tree for estimating skin irritation and corrosion potential, based on rules published in [WAL05]
4. A decision tree for estimating eye irritation and corrosion potential, based on rules published in [GER05]
5. A decision tree for estimating carcinogenicity and mutagenicity [BEN07], [BEN08]

ToxTree could be applied to datasets from various compatible file types. User-defined molecular structures are also supported – they could be entered by SMILES, or by using the built-in 2D structure diagram editor. The ToxTree has been designed with flexible capabilities for future extensions in mind (e.g. other classification schemes that could be developed at a future date). New decision trees with arbitrary rules can be built with the help of graphical user interface or by developing new plug-ins.

General Information about the service	
Service description	ToxTree modules for estimating toxic hazard of chemical compounds. Various endpoints
URI	http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreecramer http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreecramer2

<p> http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreeverhaar http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreeeye http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreeskinirritation http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreemic http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreeskinsens http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreemichaelacceptors http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreecarc http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreekroes </p>
<p>OpenTox API Reference</p> <p>www.opentox.org/dev/apis/api-1.2/Algorithm</p>
<p>Algorithm type</p> <p>Classification via expert rules, published in the peer reviewed literature. Not a machine learning algorithm.</p>
<p>Partner responsible for the implementation</p> <p>IDEA</p>
<p>Contact within OT</p> <p>nina@acad.bg</p>

Request/Response Information	
<p>Posted Parameters</p> <p>dataset_uri (mandatory)</p> <p>(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p> <p>dataset_service (optional)</p>	
<p>Response</p> <p>A task URI is provided if the service tries to calculate a regression model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.</p>	

Status Codes	
200	Success – The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect – the result can be found elsewhere.
400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://apps.ideaconsult.net:8080/ambit2/algorithm?search=ToxTree
500	Internal Server Error – The parameters you have posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it

	received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information
HTTP Method POST
Programming Language Java
Libraries used CDK, MOPAC 7.1 for the Benigni/Bossa rules for predicting carcinogenicity and mutagenicity

Examples
Example curl -X POST -d dataset_uri= http://apps.ideaconsult.net:8080/ambit2/dataset/R545 -d subjectid:YOUR-TOKEN http://apps.ideaconsult.net:8080/ambit2/algorithm/toxtreecramer

5.4.9 WEKA machine learning algorithms

General Information about the service
Service description Regression and Classification algorithms, based on Weka machine learning package. http://apps.ideaconsult.net:8080/ambit2/algorithm?type=Regression http://apps.ideaconsult.net:8080/ambit2/algorithm?type=Classification
URI http://apps.ideaconsult.net:8080/ambit2/algorithm/148 http://apps.ideaconsult.net:8080/ambit2/algorithm/RandomForest http://apps.ideaconsult.net:8080/ambit2/algorithm/Functional+tree http://apps.ideaconsult.net:8080/ambit2/algorithm/IB1 http://apps.ideaconsult.net:8080/ambit2/algorithm/MLP http://apps.ideaconsult.net:8080/ambit2/algorithm/SMO http://apps.ideaconsult.net:8080/ambit2/algorithm/Winnow http://apps.ideaconsult.net:8080/ambit2/algorithm/Bayesian+Logistic+Regression http://apps.ideaconsult.net:8080/ambit2/algorithm/DMNBtext http://apps.ideaconsult.net:8080/ambit2/algorithm/NaiveBayes http://apps.ideaconsult.net:8080/ambit2/algorithm/NBM http://apps.ideaconsult.net:8080/ambit2/algorithm/BayesianLogisticRegression http://apps.ideaconsult.net:8080/ambit2/algorithm/AODE http://apps.ideaconsult.net:8080/ambit2/algorithm/HNB http://apps.ideaconsult.net:8080/ambit2/algorithm/LR http://apps.ideaconsult.net:8080/ambit2/algorithm/GaussianProcesses http://apps.ideaconsult.net:8080/ambit2/algorithm/IsotonicRegression

http://apps.ideaconsult.net:8080/ambit2/algorithm/LMSLR http://apps.ideaconsult.net:8080/ambit2/algorithm/LogisticRegression http://apps.ideaconsult.net:8080/ambit2/algorithm/MLP http://apps.ideaconsult.net:8080/ambit2/algorithm/PaceRegresion http://apps.ideaconsult.net:8080/ambit2/algorithm/RBFNetwork http://apps.ideaconsult.net:8080/ambit2/algorithm/SMOreg http://apps.ideaconsult.net:8080/ambit2/algorithm/VotedPerceptron
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Regression, Classification
Partner responsible for the implementation Ideaconsult Ltd.
Contact within OT Nina Jeliaskova <jeliaskova.nina@gmail.com>

Request/Response Information	
Posted Parameters	
dataset_uri (mandatory) The data set URI is a mandatory parameter that has to be specified by the client. When assessing a training set of a model, this is the training set of the model.	
prediction_feature (mandatory for supervised learning algorithms, ignored by clustering algorithms)	
param (optional, algorithm specific parameters, as defined by WEKA option syntax)	
dataset_service (optional, specifies the URI of the dataset service, where the result will be stored. By default the dataset service ,where the training dataset reside is used.)	
Response	
A task, which, upon successful completion, will be pointing to the URI of the trained model, is created and returned to the client. The new model URI is then used to apply the model to a query compound or dataset.	

Status Codes	
202	Accepted - The request has succeeded and a task is returned to the client.
400	Bad Request - Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. A 400 will occur in case the prediction feature is not a subclass of ot:NumericFeature for regression algorithms, or not a subclass of ot:NominalFeature for classification algorithms.
404	The resource was not found - check your spelling.
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.

Implementation Information
HTTP Method POST
Programming Language The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://apps.ideaconsult.net:8080/ambit2 runs within an Apache Tomcat container.
Libraries used WEKA version 3.6.2. is used for the implementation of all WEKA machine learning algorithms. The full list of dependencies is available via maven pom.xml files of AMBIT project.

Examples
Example <pre>curl -X POST -d dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545 -d prediction_feautre=http://apps.ideaconsult.net:8080/ambit2/feature/22200 -H subjectid:YOUR-TOKEN http://apps.ideaconsult.net:8080/ambit2/algorithm/LR</pre>

5.4.10 Bayes Net

A Bayesian network is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

General Information about the service
Service description The BayesNet web service enables the user to build Bayes Net models for a specific data set.
URI http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/BayesNet (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/BayesNet (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Classification
Partner responsible for the implementation Technische Universität München
Contact within OT kramer@in.tum.de

Request/Response Information
Posted Parameters
dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/425254) which should be available in RDF format.

estimator (optional, default: SimpleEstimator)

The estimator algorithm to be used in the compound. Must be SimpleEstimator, MultiNomialBMAEstimator, BMAEstimator or BayesNetEstimator.

estimatorParams (optional, default: 0.5)

The parameter for the estimator to be used in the compound.

searchAlgorithm (optional, default: local.K2)

The algorithm to be used for searching in the compound. Must be local.K2, local.GeneticSearch, local.HillClimber, local.LAGDHillClimber, local.RepeatedHillClimber, local.SimulatedAnnealing, local.TabuSearch, local.TAN, global.K2, global.GeneticSearch, global.HillClimber, global.RepeatedHillClimber, global.SimulatedAnnealing, global.TabuSearch, global.TAN, ci.CISearchAlgorithm, ci.ICSSearchAlgorithm.

searchParams (optional, default: -P 1 -S BAYES -E)

The parameter for algorithm to be used for searching in the compound. Are set automatically (WEKA's standard parameter setting).

useADTree (optional, default: true)

Whether to use ADTrees for searching (using will increase the speed of the search, but will also raise the memory use).

Response

A task URI is provided if the service tries to calculate a classification model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes

200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error - The parameters you have posted are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being - Try again later!

Implementation Information
HTTP Method POST
Programming Language The project was built in Java and runs as a standalone application.
Libraries used The service uses the WEKA software package.

Examples
Example <pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/585758" -d "prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/111148" http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/BayesNet</pre>

5.4.11 Linear Regression

LR (Linear Regression) is a simple and popular statistical technique that uses explanatory (independent) variables to predict the outcome of a response (dependent) variable. The model creates a relationship in the form of a straight line (linear) that best approximates all the individual data points.

General Information about the service
Service description The linear regression web service enables the user to build linear regression models for a specific data set.
URI http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/LR (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/LR (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Regression
Partner responsible for the implementation Technische Universität München
Contact within OT kramer@in.tum.de

Request/Response Information
Posted Parameters

dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.

attributeSelectionMethod (optional, default: 1)

The attribute selection method to be used.

eliminateColinearAttributes (optional, default: 1)

Whether to eliminate colinear attributes.

ridge (optional, default: 1.0E-8)

The ridge parameter.

Response

A task URI is provided if the service tries to calculate a regression model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes

200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error - The parameters you have posted are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being - Try again later!

Implementation Information
HTTP Method

POST

Programming Language

The project was built in Java and runs as a standalone application.

Libraries used

The service uses the WEKA software package.

Examples
Example

```
curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/598242"
```

-d "prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200"
 http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/LR

5.4.12 PLS

One way to understand Partial-least squares regression (PLS) is that it simultaneously projects the x and y variables onto the same subspace in such a way that there is a good relationship between the predictor and response data. Another way to see PLS is that it forms "new" x variables as linear combinations of the old ones, and subsequently uses these new linear combinations as predictors of y . Hence, as opposed to MLR, PLS can handle correlated variables, which are noisy and possibly also incomplete.

General Information about the service	
Service description	The PLS web service enables the user to build regression models for a specific data set.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/PLSregression (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/PLSregression (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Regression
Partner responsible for the implementation	Technische Universität München
Contact within OT	kramer@in.tum.de

Request/Response Information	
Posted Parameters	<p>dataset_uri (mandatory) (e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p> <p>numComponents (optional, default: 5) The number of components to compute.</p> <p>performPrediction (optional, default: 0) Whether to update the class attribute with the predicted value.</p> <p>preprocessing (optional, default: center) Sets the type of preprocessing to use. Can be none, center or standardize.</p> <p>replaceMissing (optional, default: 0) Whether to replace missing values.</p>

Response

A task URI is provided if the service tries to calculate a regression model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes

200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being - Try again later!

Implementation Information
HTTP Method

POST

Programming Language

The project was built in Java and runs as a standalone application.

Libraries used

The service uses the WEKA software package.

Examples
Example 1

```
curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/598242"
-d "prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200"
http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/PLSregression
```

5.4.13 LoMoGraph

The algorithm combines clustering and classification or regression for making predictions on chemical structure data. A clustering procedure producing clusters with shared structural scaffolds is applied as a preprocessing step, before a (local) model is learned for each relevant cluster. Instead of using only one global model (classical approach), LoMoGraph uses weighted local models for predictions of query compounds dependent on cluster memberships. Thus, LoMoGraph is an interplay of structural clustering, model learning and prediction.

General Information about the service	
Service description	The LoMoGraph web service enables the user to build regression or classification models, depending on the data set.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/LoMoGraphRegression (AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/LoMoGraphClassification (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Regression, Classification
Partner responsible for the implementation	Technische Universität München
Contact within OT	kramer@in.tum.de

Request/Response Information	
Posted Parameters	<p>dataset_uri (mandatory) (e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p> <p>threshold (optional, default: 0.4) The fraction to which molecules must overlap to be part of the same cluster.</p> <p>minimumClusterSize (optional, default: 5) The minimum size a cluster must have so that a local model is learned out of it.</p> <p>weka_parameter_string (optional, default: -W weka.classifiers.lazy.IBk) The parameters for the model learning procedure (in WEKA command line style).</p>
Response	A task URI is provided if the service tries to calculate a model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes	
200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.

400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
401	Unauthorized – The user is not authorized to perform the underlying operation or the user did not provide a valid token ID. It has been made easy to acquire a token using our online form at http://opentox.ntua.gr:8080/login – provide your credentials of (in case you don't have an account use the username <i>guest</i> and the same password).
404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information
HTTP Method POST
Programming Language The project was built in Java and runs as a standalone application.
Libraries used The service uses the WEKA software package and a modified version of gSpan'.

Examples
Example <pre>curl -i -X POST -d 'dataset_service=http://apps.ideaconsult.net:8080/ambit2/dataset' -d 'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/598242' -d 'prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200' -H 'subjectid:AQIC5wM2LY4SfcyKFC9MJEDMigp0fO7EXIHULTEJx4cIRLY=@AAJTSQACMDE=#' -d 'threshold=0.70' -d 'minimumClusterSize=20' -d 'weka_parameter_string=-W weka.classifiers.functions.LinearRegression -S 1 -C -R 1.0E-8' http://opentox.informatik.tu-muenchen.de:8080/OpenTox- dev/algorithm/LoMoGraphRegression</pre>

5.4.14 Interval Estimators

To quantify uncertainty in QSAR prediction, the conditional density of activity, given the structure, instead of a point estimate can be used. Using a conditional density estimate, prediction intervals of activities can be derived. Three types of conditional density or interval estimators are provided: Histogram, normal and kernel estimators. These are based on generic machine learning algorithms so that an arbitrary number of attributes can be used to determine a conditional density estimate.

General Information about the service
Service description

The interval estimator web service enables the user to predict intervals for test instances.	
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/IntervalEstimator (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/IntervalEstimator (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Interval Estimator
Partner responsible for the implementation	Technische Universität München
Contact within OT	kramer@in.tum.de

Request/Response Information

Posted Parameters

dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.

baseClassifier (optional, default: RandomCommittee)

The base classifier to be used. Must be RandomCommittee or J48 (Default: RandomCommittee).

deleteEmptyBins (optional, default: 0)

Whether to delete empty bins after discretization.

estimatorType (optional, default: 0)

The density estimator to use. Must be histogram (0), kernel (1) or normal (2) estimator.

numBins (optional, default: 10)

Number of bins for the discretization.

useEqualFrequencyDiscretization (optional, default: 0)

If set to true, equal-frequency binning will be used instead of equal-width binning.

Response

A task URI is provided if the service tries to calculate a model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes

200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.

400	Bad Request – Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found – Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information	
HTTP Method	POST
Programming Language	The project was built in Java and runs as a standalone application.
Libraries used	The service uses the WEKA software package.

Examples	
Example	<pre>curl -i -X POST -d 'dataset_service=http://apps.ideaconsult.net:8080/ambit2/dataset' -d 'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/598242' -d 'prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200' -d 'estimatorType=1' http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/IntervalEstimator</pre>

5.4.15 iSAR

iSAR (instance-based structure-activity relationships) is an implementation of a lazy SAR algorithm. In lazySARs, classifications are particularly tailored for each test compound. Therefore, it is possible to make the most of the structure of a test compound. iSAR uses subgraphs and paths that are generated by e.g., Free Tree Miner, as features for the classification task. These substructures are derived from a test compound to determine similar structures. In order to obtain a well-balanced and representative set of structural descriptors, this set can be enriched by strongly activating or deactivating fragments from the training set and subsequently redundant fragments (use only closed features) can be removed. Finally, a k -Nearest Neighbour classification with one k or for several values of k is performed and a vote among the resulting predictions is taken. The validation is performed via leave-one-out cross validation (LOOCV).

General Information about the service	
Service description	The iSAR web service enables the user to build classification models for a specific data set.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/iSAR (AA)

OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Classification
Partner responsible for the implementation Technische Universität München
Contact within OT kramer@in.tum.de

Request/Response Information
Posted Parameters <p>dataset_uri (mandatory) (e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/425254) which should be available in RDF format.</p> <p>minSup (optional, default: 0.7) Specifies the minimum support for mining.</p> <p>hydrogen (optional, default: 0) Include hydrogen atoms.</p> <p>closed (optional, default: 0) Use only closed features.</p>
Response <p>A task URI is provided if the service tries to calculate a classification model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.</p>

Status Codes	
200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being - Try again later!

Implementation Information
HTTP Method POST
Programming Language The project was built in Java and runs as a standalone application.
Libraries used The service uses the FTM, JOELib2 and WEKA software packages.

Examples
Example <pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/585758" -d "prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/111148" http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/iSAR</pre>

5.4.16 Multiple Linear Regression

MLR (Multiple Linear Regression) is a simple and popular statistical technique that uses several explanatory (independent) variables to predict the outcome of a response (dependent) variable. The model creates a relationship in the form of a straight line (linear) that best approximates all the individual data points.

General Information about the service
Service description Training algorithm for multiple linear regression models. MLR is applied to data sets which contain exclusively numeric data entries.
URI http://opentox.ntua.gr:8080/algorithm/mlr (AA)
OpenTox API Reference http://www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Regression, Single Target
Partner responsible for the implementation National Technical University of Athens, School of Chemical Engineering, Automatic Control Unit.
Contact within OT Pantelis Sopasakis < chvng@mail.ntua.gr >

Request/Response Information
Posted Parameters dataset_uri (mandatory) As every model training algorithm, the data set URI is a mandatory parameter that has to be specified

by the client. Unless it is a valid data set URI, a status code 400 is returned. (e.g. <http://apps.ideaconsult.net:8080/ambit2/dataset/R545>). It is highly recommended that the submitted data set has no missing values otherwise a Missing Value Resolver will run over the input set of values leading in models of ambiguous quality.

prediction_feature (mandatory)

A feature among the ones in the data set submitted to the service as dataset_uri. Necessarily must be of type 'Numeric'.

Response

A task, which, upon successful completion, will be pointing to the URI of the trained model, is created and returned to the client.

Status Codes

202	Accepted – The request has succeeded and a task is returned to the client.
400	Bad Request – Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. Check under /dataset/id/feature for a list of features available. A 400 will also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled).
401	Unauthorized – The user is not authorized to perform the underlying operation or the user did not provide a valid token ID. It has been made easy to acquire a token using our online form at http://opentox.ntua.gr:8080/login – provide your credentials of (in case you don't have an account use the username <i>guest</i> and the same password).
404	The resource was not found – check your spelling.
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down. Troubleshooting is facilitated as in such a case an <i>Error Report</i> is returned to the client.
503	The service is not available for the time being – Try again later!
507	Insufficient storage – The user has exceeded their quota. Check at /user/id/quota for details. Mail the system administrator if you need more space on the server. By default, all users are allowed for 2000 models, 2000 BibTeX entries and 5 tasks running in parallel. The problem can be resolved if old models are deleted. Note that after a deletion, the models will be still on the server for 30days (but no guarantee is provided) ; read http://opentox.ntua.gr/index.php/blog/69-oops-i-deleted-my-favorite-model for details.

Implementation Information

HTTP Method

POST

Programming Language

The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at <http://opentox.ntua.gr:8080> runs within an Apache Tomcat

container.

Libraries used

WEKA v. 3.6.0 was used for the implementation of this particular algorithm. For a list of all dependencies of the JAQPOT3 project (<https://github.com/alphaville/jaqpot3>) read <https://github.com/alphaville/jaqpot3/blob/master/jaqpot3-standalone/pom.xml> .

Examples

Example

```
curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -d
"prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200" -H subjectid:YOUR-TOKEN
http://opentox.ntua.gr:8080/algorithm/mlr
```

5.4.17 Support Vector Machine

Support vector machines (SVM) are a set of supervised learning methods used for classification and regression. In the most widely used two-class SVM classification method, input data are viewed as two sets of vectors in the multi-dimensional input space. The SVM classifier constructs a separating hyperplane in that space, one which maximizes the margin between the two data sets. The method is extended to multi-class and nonlinear classification problems by using nonlinear kernel function. To obtain an optimum classifier for non-separable data, a penalty is introduced for misclassified data. This penalty is zero for patterns classified correctly, and has a positive value that increases with the distance from the corresponding hyperplane for patterns that are not situated on the correct side of the classifier. Similar concepts are used in the SVM regression problem, where the objective is to identify a function that for all training patterns has a maximum deviation ϵ from the target (experimental) values.

General Information about the service

Service description

Algorithm for training regression models using the Support Vector Machine Learning Algorithm. The training is based on the WEKA implementation of SVM and specifically the class `weka.classifiers.functions.SVMreg`. A comprehensive introductory text is provided by John Shawe-Taylor and Nello Cristianini in the book 'Support Vector Machines' Cambridge University Press, 2000

URI

<http://opentox.ntua.gr:8080/algorithm/svm> (AA)

OpenTox API Reference

www.opentox.org/dev/apis/api-1.2/Algorithm

Algorithm type

Regression, Single Target

Partner responsible for the implementation

National Technical University of Athens, School of Chemical Engineering, Automatic Control Unit.

Contact within OT

Pantelis Sopasakis <chvng@mail.ntua.gr>

Request/Response Information	
Posted Parameters	
dataset_uri (mandatory)	
<p>As every model training algorithm, the data set URI is a mandatory parameter that has to be specified by the client. Unless it is a valid data set URI, a status code 400 is returned. (e.g. http://apps.ideaconsult.net:8080/ambit2/dataset/R545). It is highly recommended that the submitted data set has no missing values otherwise a Missing Value Resolver will run over the input set of values leading in models of ambiguous quality.</p>	
prediction_feature (mandatory)	
<p>A feature among the ones in the data set submitted to the service as dataset_uri. Necessarily must be of type 'Numeric'.</p>	
epsilon (optional, default=0.1)	
<p>The ϵ parameter of the SVM algorithm. Only strictly positive values are allowed.</p>	
kernel (optional, default=RBF)	
<p>The kernel function used for the SVM model. Linear, Polynomial and RBF kernels are available. Admissible values are 'RBF', 'linear' and 'polynomial'.</p>	
tolerance (optional, default=0.0001)	
<p>The tolerance used as a termination criterion in the training procedure. Lowering this value below the default will lead to more accurate training but might significantly increase the computational time or even lead to infeasibility (the algorithm might fail to terminate). Only strictly positive values are allowed and we advise users to use values that do not exceed 0.10</p>	
gamma (optional, default=1.5)	
<p>The gamma parameter for the RBF kernel. Only strictly positive values are allowed here. The choice of this parameter is critical for the performance of the model.</p>	
cost (optional, default=100)	
<p>The C parameter involved in the cost function used for the training of the SVM model.</p>	
Response	
<p>A task, which, upon successful completion, will be pointing to the URI of the trained model, is created and returned to the client.</p>	

Status Codes	
202	Accepted - The request has succeeded and a task is returned to the client.
400	Bad Request - Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. Check under /dataset/id/feature for a list of features available. A 400 will also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled). Invalid parametrization can also occur if optional parameters are set to non-meaningful values (e.g. a negative value of epsilon).
401	Unauthorized - The user is not authorized to perform the underlying operation or the user did not provide a valid token ID. It has been made easy to acquire a token using our online form at http://opentox.ntua.gr:8080/login - provide your credentials of (in case you don't have an account

	use the username <i>guest</i> and the same password).
404	The resource was not found – check your spelling.
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down. Troubleshooting is facilitated as in such a case an <i>Error Report</i> is returned to the client.
503	The service is not available for the time being – Try again later!
507	Insufficient storage – The user has exceeded their quota. Check at <code>/user/id/quota</code> for details. Mail the system administrator if you need more space on the server. By default, all users are allowed for 2000 models, 2000 BibTeX entries and 5 tasks running in parallel. The problem can be resolved if old models are deleted. Note that after a deletion, the models will be still on the server for 30days (but no guarantee is provided) ; read http://opentox.ntua.gr/index.php/blog/69-oops-i-deleted-my-favorite-model for details.

Implementation Information

HTTP Method

POST

Programming Language

The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at <http://opentox.ntua.gr:8080> runs within an Apache Tomcat container.

Libraries used

WEKA v. 3.6.0 was used for the implementation of this particular algorithm. For a list of all dependencies of the JAQPOT3 project (<https://github.com/alphaville/jagpot3>) read <https://github.com/alphaville/jagpot3/blob/master/jagpot3-standalone/pom.xml> .

Examples

Example

```
curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -d
"prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200" -d gamma=0.76 -H
subjectid:YOUR-TOKEN http://opentox.ntua.gr:8080/algorithm/svm
```

5.4.18 Radial Basis Function Neural Network

Fast-RBF-NN is a training algorithm for Radial Basis Function Neural Networks. The algorithm is based on the subtractive clustering technique and has a number of advantages compared to the traditional learning algorithms including faster training times and more accurate predictions. Due to these advantages the method proves suitable for developing models for complex nonlinear systems. The algorithm is presented in detail in [SAR2003]

General Information about the service

Service description A Fast implementation of the RBF NN algorithm, as described in [SAR2003] has been implemented as an OpenTox API v.1.2. compliant web service and has been deployed at http://opentox.ntua.gr:8080/algorithm/fastRbfNn .
URI http://opentox.ntua.gr:8080/algorithm/fastRbfNn (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Regression, Single Target
Partner responsible for the implementation National Technical University of Athens, School of Chemical Engineering, Automatic Control Unit.
Contact within OT Pantelis Sopasakis < chvng@mail.ntua.gr >

Request/Response Information
Posted Parameters <p>dataset_uri (mandatory)</p> <p>As every model training algorithm, the data set URI is a mandatory parameter that has to be specified by the client. Unless it is a valid data set URI, a status code 400 is returned. (e.g. http://apps.ideaconsult.net:8080/ambit2/dataset/R545). It is highly recommended that the submitted data set has no missing values otherwise a Missing Value Resolver will run over the input set of values leading in models of ambiguous quality.</p> <p>prediction_feature (mandatory)</p> <p>A feature among the ones in the data set submitted to the service as dataset_uri. Necessarily must be of type 'Numeric'.</p> <p>a (optional, default=1.0)</p> <p>Design parameter involved in the calculation of the initial potential of all vectors of the training set according to the formula $P(i)=\sum_{j=1}^K \exp(-a* x(i)-x(j) ^2)$ for $i=1,2,\dots,K$.</p> <p>b (optional, default=0.9)</p> <p>A design parameter that is suggested to be chosen smaller than a to avoid the selection of closely located hidden nodes. This parameter is involved in the formula that defines the potential update in every step of the algorithm, that is $P(i) = P(i) - P(L)\exp(x(i)-x^*(L) ^2)$.</p> <p>e (optional, default=0.6)</p> <p>Parameter used to implicitly determine the number of iterations and therefore the number hidden nodes the algorithm will find. The algorithm terminates when $\max_i P(i)$ is less than or equal to $e*P^*(L)$</p>
Response A task, which, upon successful completion, will be pointing to the URI of the trained model, is created

and returned to the client.

Status Codes	
202	Accepted – The request has succeeded and a task is returned to the client.
400	Bad Request – Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. Check under /dataset/id/feature for a list of features available. A 400 will also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled). Invalid parametrization can also occur if optional parameters are set to non-meaningful values (e.g. a negative value of a, b, or e).
401	Unauthorized – The user is not authorized to perform the underlying operation or the user did not provide a valid token ID. It has been made easy to acquire a token using our online form at http://opentox.ntua.gr:8080/login – provide your credentials of (in case you don't have an account use the username <i>guest</i> and the same password).
404	The resource was not found – check your spelling.
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down. Troubleshooting is facilitated as in such a case an <i>Error Report</i> is returned to the client.
503	The service is not available for the time being – Try again later!
507	Insufficient storage – The user has exceeded their quota. Check at /user/id/quota for details. Mail the system administrator if you need more space on the server. By default, all users are allowed for 2000 models, 2000 BibTeX entries and 5 tasks running in parallel. The problem can be resolved if old models are deleted. Note that after a deletion, the models will be still on the server for 30 days (but no guarantee is provided) ; read http://opentox.ntua.gr/index.php/blog/69-oops-i-deleted-my-favorite-model for details.

Implementation Information
<p>HTTP Method</p> <p>POST</p>
<p>Programming Language</p> <p>The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://opentox.ntua.gr:8080 runs within an Apache Tomcat container.</p>
<p>Libraries used</p> <p>JAMA was used for the implementation of this particular algorithm. For a list of all dependencies of the JAQPOT3 project (https://github.com/alphaville/jaqpot3) read https://github.com/alphaville/jaqpot3/blob/master/jaqpot3-standalone/pom.xml .</p>

Examples

Example

```
curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -d
"prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200" -d a=0.5 -d b=0.45 -d e=0.8
-H subjectid:YOUR-TOKEN http://opentox.ntua.gr:8080/algorithm/fastRbfNn
```

5.4.19 MaxTox

MaxTox provides a service to generate fingerprints for a given set of compounds using a Maximum Common Substructure Search (MCSS) approach which can be used for making prediction models using a machine learning algorithms like SVM. The MaxTox service is available at <http://202.141.146.74:8080/MaxtoxMCSS> and <http://opentox2.informatik.uni-freiburg.de:8080/MaxtoxMCSS>. It is a standalone service that can also be used to generate a MCSS dictionary for a given set of compounds and then to create a prediction model using SVMs. Also, these fingerprints can be fed to the models stored at the MaxTox server to predict the toxicity of the query compound. MaxTox works with the OpenTox API 1.2 specification and can consume and respond in RDF data types. The MaxTox service can be used as a component of other prediction use-cases hosted from other servers, as long as the data transactions are performed according to the OpenTox API.

General Information about the service	
Service description	MaxTox web service enables the user to generate fingerprints as descriptors using MaxTox algorithm, build the prediction model and predict the toxicity of the unknown compounds.
URI	http://opentox2.informatik.uni-freiburg.de:8080/MaxtoxMCSS (non-AA) http://202.141.146.74:8080/MaxtoxMCSS (non-AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Classification
Partner responsible for the implementation	SL-JNU
Contact within OT	indirag@mail.jnu.ac.in

Request/Response Information
Posted Parameters
dataset_uri (mandatory)

(e.g. dataset_uri= <http://apps.ideaconsult.net:8080/ambit2/dataset/425254>) which should be available in RDF format.

Response

A task URI is provided if the service tries to calculate a classification model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes	
200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, received an unsuccessful response. In those cases, it seems that some other server is down
503	The service is not available for the time - Try again later!

Implementation Information
HTTP Method POST
Programming Language The project was built in Java and runs as a standalone application.
Libraries used The service uses the CDK software package.

Examples
Example 1 (non-AA) : for creating fingerprints <pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/272?max=2" http://opentox2.informatik.uni-freiburg.de:8080/MaxtoxMCSS/algorithm/MCSSFinder</pre>
Example 2 (non-AA) : for predicting toxicity of a data set <pre>curl -X POST -H "Content-Type:application/x-www-form-urlencoded" -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/272?max=2" http://opentox2.informatik.uni-freiburg.de:8080/MaxtoxMCSS/model/<model_id></pre>

5.5 Clustering Algorithms

5.5.1 Structural Clustering

TUM's structural clustering procedure works on structural graph data, without generating features or decomposing graphs into parts. In contrast to many related approaches, the method does not rely on computationally expensive maximum common subgraph (MCS) operations or variants thereof, but on frequent subgraph mining. More specifically, the problem formulation takes advantage of the frequent subgraph miner gSpan (that performs well on many practical problems) without effectively generating thousands of subgraphs in the process. In the proposed clustering approach, clusters encompass all graphs that share a sufficiently

large common subgraph. The size of the common subgraph of a graph in a cluster has to take at least a user-specified fraction of its overall size. The structural clustering procedure works in an online mode (processing one structure after the other) and produces overlapping (non-disjoint) and non-exhaustive clusters.

General Information about the service	
Service description	The structural clustering web service enables the user to cluster chemical structure data.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/StructuralClustering (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/StructuralClustering (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Clustering
Partner responsible for the implementation	Technische Universität München
Contact within OT	kramer@in.tum.de

Request/Response Information	
Posted Parameters	<p>dataset_uri (mandatory)</p> <p>(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p> <p>threshold (optional, default: 0.4)</p> <p>The fraction to which molecules must overlap to be part of the same cluster.</p>
Response	A task URI is provided if the service tries to calculate a clustering model. The task URI can be queried (GET) for the resulting model URI. Otherwise an explanatory message is provided.

Status Codes	
200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
303	Redirect - the result can be found elsewhere.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms,

	check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information	
HTTP Method	POST
Programming Language	The project was built in Java and runs as a standalone application.
Libraries used	The service uses a modified version of gSpan'.

Examples	
Example 1	<pre>curl -i -X POST -d 'dataset_service=http://apps.ideaconsult.net:8080/ambit2/dataset' -d 'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/3553' -d 'threshold=0.4' http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/StructuralClustering</pre>

5.6 Feature Selection, Data Transformation and Filtering Algorithms

5.6.1 Information Gain Attribute Evaluation

InfoGainAttributeEval evaluates the worth of an attribute by measuring the information gain with respect to the class.

$$\text{InfoGain}(\text{Class}, \text{Attribute}) = H(\text{Class}) - H(\text{Class} \mid \text{Attribute}),$$

where H is the information entropy.

General Information about the service	
Service description	This service evaluates the worth of an attribute by measuring the information gain with respect to the class.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/InfoGainAttributeEval (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/InfoGainAttributeEval (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm

Algorithm type Feature Selection
Partner responsible for the implementation Technische Universität München
Contact within OT kramer@in.tum.de

Request/Response Information
Posted Parameters <p>dataset_uri (mandatory) (e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.</p> <p>binarizeNumericAttributes (optional, default: 0) Just binarize numeric attributes instead of properly discretizing them.</p> <p>missingMerge (optional, default: 1) Distribute counts for missing values. Counts are distributed across other values in proportion to their frequency. Otherwise, missing is treated as a separate value.</p> <p>numToSelect (optional, default: -1) Specify the number of attributes to retain. The default value (-1) indicates that all attributes are to be retained. Use either this option or a threshold to reduce the attribute set.</p> <p>startSet (optional, default: empty) Specify a set of attributes to ignore. When generating the ranking, Ranker will not evaluate the attributes in this list. This is specified as a comma-separated list of attribute indexes starting at 1. It can include ranges e.g. 1,2,5-9,17.</p> <p>threshold (optional, default: -1.7976931348623157E308) Set threshold by which attributes can be discarded. Default value results in no attributes being discarded. Use either this option or numToSelect to reduce the attribute set.</p>
Response A task URI is provided if the service selects features. The task URI can be queried (GET) for the resulting dataset URI. Otherwise an explanatory message is provided.

Status Codes	
200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/

500	Internal Server Error – The parameters you posted are acceptable but some internal error occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being – Try again later!

Implementation Information	
HTTP Method	POST
Programming Language	The project was built in Java and runs as a standalone application.
Libraries used	The service uses the class InfoGainAttributeEval which is included in WEKA (version 3.6.0).

Examples	
Example	<pre>curl -X POST -d 'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/585758' -d 'prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/111148' -d 'numToSelect=10' http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/InfoGainAttributeEval -iv</pre>

5.6.2 Chi Squared Attribute Evaluation

Feature Selection via chi square (X^2) test is another, very commonly used method. The X^2 method evaluates features individually by measuring their chi-squared statistic with respect to the classes.

General Information about the service	
Service description	Evaluates the worth of an attribute by computing the value of the Chi-Squared statistic with respect to the class.
URI	http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/ChiSquared (non-AA) http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/ChiSquared (AA)
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type	Feature Selection
Partner responsible for the implementation	Technische Universität München
Contact within OT	kramer@in.tum.de

Request/Response Information

Posted Parameters

dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.

binarizeNumericAttributes (optional, default: 0)

Just binarize numeric attributes instead of properly discretizing them.

missingMerge (optional, default: 1)

Distribute counts for missing values. Counts are distributed across other values in proportion to their frequency. Otherwise, missing is treated as a separate value.

numToSelect (optional, default: -1)

Specify the number of attributes to retain. The default value (-1) indicates that all attributes are to be retained. Use either this option or a threshold to reduce the attribute set.

startSet (optional, default: empty)

Specify a set of attributes to ignore. When generating the ranking, Ranker will not evaluate the attributes in this list. This is specified as a comma-separated list of attribute indexes starting at 1. It can include ranges e.g. 1,2,5-9,17.

threshold (optional, default: -1.7976931348623157E308)

Set threshold by which attributes can be discarded. Default value results in no attributes being discarded. Use either this option or numToSelect to reduce the attribute set.

Response

A task URI is provided if the service selects features. The task URI can be queried (GET) for the resulting dataset URI. Otherwise an explanatory message is provided.

Status Codes

200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error - The parameters you posted are acceptable but some internal error occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being - Try again later!

Implementation Information

HTTP Method

POST
Programming Language
The project was built in Java and runs as a standalone application.
Libraries used
The service uses the class ChiSquared which is included in WEKA (version 3.6.0).

Examples
Example
<pre>curl -X POST -d 'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/585758' -d 'prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/111148' -d 'numToSelect=10' http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/ChiSquared -iv</pre>

5.6.3 PCA

The Principle Component Analysis (PCA) is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate and so forth. The coordinates are here called principal components.

General Information about the service
Service description
The PCA web service performs a principal components analysis and transformation of the data.
URI
http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/PrincipalComponents (non-AA)
http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/algorithm/PrincipalComponents (AA)
OpenTox API Reference
www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type
Feature Selection
Partner responsible for the implementation
Technische Universität München
Contact within OT
kramer@in.tum.de

Request/Response Information
Posted Parameters
dataset_uri (mandatory)

(e.g. dataset_uri=ambit.uni-plovdiv.bg:8080/ambit2/dataset/R545) which should be available in RDF format.

maximumAttributeNames (optional, default: 5)

The maximum number of attributes to include in transformed attribute names.

normalize (optional, default: 1)

Whether to normalize the input.

transformBacktoOriginal (optional, default: 0)

Transform through the PC space and back to the original space. If only the best n PCs are retained (by setting varianceCovered < 1) then this option will give a data set in the original space but with less attribute noise.

varianceCovered (optional, default: 0.95)

Retain enough PC attributes to account for this proportion of variance.

numToSelect (optional, default: -1)

Specify the number of attributes to retain. The default value (-1) indicates that all attributes are to be retained. Use either this option or a threshold to reduce the attribute set.

startSet (optional, default: empty)

Specify a set of attributes to ignore. When generating the ranking, Ranker will not evaluate the attributes in this list. This is specified as a comma-separated list of attribute indexes starting at 1. It can include ranges e.g. 1,2,5-9,17.

threshold (optional, default: -1.7976931348623157E308)

Set threshold by which attributes can be discarded. Default value results in no attributes being discarded. Use either this option or numToSelect to reduce the attribute set.

Response

A task URI is provided if the service selects features. The task URI can be queried (GET) for the resulting dataset URI. Otherwise an explanatory message is provided.

Status Codes

200	Success - The request has succeeded and the requested features were generated. The URI of the data set is returned within the response body.
400	Bad Request - Some parameter you provided is wrong or you did not post some mandatory parameter such as the dataset_uri.
404	The resource was not found - Check your spelling. For a complete list of all available algorithms, check out http://opentox.informatik.tu-muenchen.de:8080/OpenTox-dev/
500	Internal Server Error - The parameters you posted are acceptable but some internal error occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down.
503	The service is not available for the time being - Try again later!

Implementation Information
HTTP Method POST
Programming Language The project was built in Java and runs as a standalone application.
Libraries used The service uses the class PrincipalComponents which is included in WEKA (version 3.6.0).

Examples
Example <pre>curl -X POST -d 'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/585758' -d 'prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/111148' http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/PrincipalComponents</pre>

5.6.4 Partial Least Squares Filter

PLS Filter is a supervised feature selection algorithm that leads to feature transformation. This transformation is encoded as an OpenTox model which has no dependent variables. PLS is a standard, widely used supervised algorithm for dimension reduction on data sets.

General Information about the service
Service description Partial Least Squares Filter (PLS) has been implemented as an OpenTox API version 1.2. compatible web service. The implementation is based on WEKA version 3.6.0 and in particular the class <code>weka.filters.supervised.attribute.PLSFilter</code> was used.
URI http://opentox.ntua.gr:8080/algorithm/pls (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Preprocessing, Supervised
Partner responsible for the implementation National Technical University of Athens, School of Chemical Engineering, Automatic Control Unit.
Contact within OT Pantelis Sopasakis < chvng@mail.ntua.gr >

Request/Response Information
Posted Parameters dataset_uri (mandatory)

As every model training algorithm, the data set URI is a mandatory parameter that has to be specified by the client. Unless it is a valid data set URI, a status code 400 is returned. (e.g. <http://apps.ideaconsult.net:8080/ambit2/dataset/R545>). It is highly recommended that the submitted data set has no missing values otherwise a Missing Value Resolver will run over the input set of values leading in models of ambiguous quality.

target (mandatory)

A feature among the ones in the data set submitted to the service as `dataset_uri` which stands as the target feature with respect to which PLS training is carried out. Necessarily must be of type 'Numeric'. This is different from the parameter `prediction_feature`. This is not kind of a dependent feature since this is a filtering algorithm and it does not generate a predictive model but transforms the submitted data set.

doUpdateClass (optional, default=off)

Whether the target feature should be updated. The target feature is specified using the mandatory parameter 'target'. Admissible values are 'on' and 'off'. Default is 'off'.

numComponents (mandatory)

The maximum number of attributes(features) to use. The number of components must be less than the number of independent features in the data set .

preprocessing (optional, default=center)

Preprocessing on the provided data prior to the application of the PLS algorithm. Admissible values are 'none', 'center' and 'standardize'.

algorithm (optional, default=PLS1)

The type of algorithm to use for the training of the PLS model; admissible values are PLS1 and SIMPLS.

Response

A task is created and returned to the client, which upon successful completion will be pointing to the URI of the data set.

Status Codes

202	Accepted - The request has succeeded and a task is returned to the client.
400	Bad Request - Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. Check under <code>/dataset/id/feature</code> for a list of features available. A 400 will be also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled). Invalid parametrization can also occur if optional parameters are set to non-meaningful values.
401	Unauthorized - The user is not authorized to perform the underlying operation or the user did not provide a valid token ID. It has been made easy to acquire a token using our online form at http://opentox.ntua.gr:8080/login - provide your credentials of (in case you don't have an account use the username <i>guest</i> and the same password).
404	The resource was not found - check your spelling.
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.

502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down. Troubleshooting is facilitated as in such a case an <i>Error Report</i> is returned to the client.
503	The service is not available for the time being – Try again later!
507	Insufficient storage – The user has exceeded their quota. Check at /user/id/quota for details. Mail the system administrator if you need more space on the server. By default, all users are allowed for 2000 models, 2000 BibTeX entries and 5 tasks running in parallel. The problem can be resolved if old models are deleted. Note that after a deletion, the models will be still on the server for 30 days (but no guarantee is provided) ; read http://opentox.ntua.gr/index.php/blog/69-oops-i-deleted-my-favorite-model for details.

Implementation Information	
HTTP Method	POST
Programming Language	The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://opentox.ntua.gr:8080 runs within an Apache Tomcat container.
Libraries used	WEKA v. 3.6.0 was used for the implementation of this particular algorithm. For a list of all dependencies of the JAQPOT3 project (https://github.com/alphaville/jaqpot3) read https://github.com/alphaville/jaqpot3/blob/master/jaqpot3-standalone/pom.xml .

Examples	
Example	<pre>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -d "target=http://apps.ideaconsult.net:8080/ambit2/feature/22200" -d numComponents=2 -d doUpdateClass=off -d algorithm=PLS1 -d preprocessing=center -H subjectid:YOUR-TOKEN http://opentox.ntua.gr:8080/algorithm/pls</pre>

5.6.5 Scaling Filter

This web service is intended to scale the numeric values of an OpenTox data set within a specified range. If not otherwise specified by the client, this range is assumed to be $[-1, 1]$. Scaling is necessary for algorithms like SVM and Neural Networks as it substantially improves the accuracy of the trained models. In other cases such as MLR it can numerically stabilize the training procedure and is one of the fundamental preprocessing steps.

General Information about the service	
Service description	A scaling filter has been implemented as an OpenTox API version 1.2. compatible web service.
URI	http://opentox.ntua.gr:8080/algorithm/scaling (AA)

OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Preprocessing
Partner responsible for the implementation National Technical University of Athens, School of Chemical Engineering, Automatic Control Unit.
Contact within OT Pantelis Sopasakis < chvng@mail.ntua.gr >

Request/Response Information	
Posted Parameters	
dataset_uri (mandatory) As every model training algorithm, the data set URI is a mandatory parameter that has to be specified by the client. Unless it is a valid data set URI, a status code 400 is returned. (e.g. http://apps.ideaconsult.net:8080/ambit2/dataset/R545). It is highly recommended that the submitted data set has no missing values otherwise a Missing Value Resolver will run over the input set of values leading in models of ambiguous quality.	
min (optional, default=0) Minimum value for the scaled data.	
max (optional, default=1) Maximum value for the scaled data.	
ignore_uri (optional) If a data set URI is provided, then the scaling is carried out with respect to the minimum and maximum values of the features in that data set. Used for applying a data set on a model that requires scaled data.	
Response A task, which, upon successful completion, will be pointing to the URI of the trained model, is created and returned to the client.	

Status Codes	
202	Accepted - The request has succeeded and a task is returned to the client.
400	Bad Request - Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. Check under /dataset/id/feature for a list of features available. A 400 will be also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled). Invalid parametrization can also occur if optional parameters are set to non-meaningful values.
401	Unauthorized - The user is not authorized to perform the underlying operation or the user did not provide a valid token ID. It has been made easy to acquire a token using our online form at http://opentox.ntua.gr:8080/login - provide your credentials of (in case you don't have an account

	use the username <i>guest</i> and the same password).
404	The resource was not found – check your spelling.
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down. Troubleshooting is facilitated as in such a case an <i>Error Report</i> is returned to the client.
503	The service is not available for the time being – Try again later!
507	Insufficient storage – The user has exceeded their quota. Check at <code>/user/id/quota</code> for details. Mail the system administrator if you need more space on the server. By default, all users are allowed for 2000 models, 2000 BibTeX entries and 5 tasks running in parallel. The problem can be resolved if old models are deleted. Note that after a deletion, the models will be still on the server for 30 days (but no guarantee is provided) ; read http://opentox.ntua.gr/index.php/blog/69-oops-i-deleted-my-favorite-model for details.

Implementation Information

HTTP Method

POST

Programming Language

The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at <http://opentox.ntua.gr:8080> runs within an Apache Tomcat container.

Libraries used

For a list of all dependencies of the JAQPOT3 project (<https://github.com/alphaville/jaqpot3>) read <https://github.com/alphaville/jaqpot3/blob/master/jaqpot3-standalone/pom.xml> .

Examples

Example

```
curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -d min=-1 -d max=1 -H subjectid:YOUR-TOKEN http://opentox.ntua.gr:8080/algorithm/scaling
```

5.6.6 Missing Values Replacer

Missing Value Handler (MVH) is an algorithm used to replace all missing values within a data set with their corresponding *means and models*. It applies only to numeric and nominal values. This action will definitely have an effect on the reliability of any model created with the data set as these values are actually 'guessed' and might strongly divert from the actual ones.

General Information about the service

Service description

Replaces missing values in the data set with new ones, leading to a dense data set using the means-and-modes approach. MVH was implemented as a web service compliant to the OpenTox API version

1.2.
URI http://opentox.ntua.gr:8080/algorithm/mvh (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Preprocessing
Partner responsible for the implementation National Technical University of Athens, School of Chemical Engineering, Automatic Control Unit.
Contact within OT Pantelis Sopasakis < chvng@mail.ntua.gr >

Request/Response Information	
Posted Parameters	
dataset_uri (mandatory)	<p>As every model training algorithm, the data set URI is a mandatory parameter that has to be specified by the client. Unless it is a valid data set URI, a status code 400 is returned. (e.g. http://apps.ideaconsult.net:8080/ambit2/dataset/R545).</p>
ignoreUri (optional)	<p>This parameter specifies a list of feature URI (among those within the data set) to be excluded from the MVH procedure. Any missing values for these features will not be replaced.</p>
Response	
<p>A task, which, upon successful completion, will be pointing to the URI of the trained model, is created and returned to the client.</p>	

Status Codes	
202	Accepted – The request has succeeded and a task is returned to the client.
400	Bad Request – Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. Check under /dataset/id/feature for a list of features available. A 400 will be also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled).
401	Unauthorized – The user is not authorized to perform the underlying operation or the user did not provide a valid token ID. It has been made easy to acquire a token using our online form at http://opentox.ntua.gr:8080/login – provide your credentials of (in case you don't have an account use the username <i>guest</i> and the same password).
404	The resource was not found – check your spelling.
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.

502	Bad Gateway – The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down. Troubleshooting is facilitated as in such a case an <i>Error Report</i> is returned to the client.
503	The service is not available for the time being – Try again later!
507	Insufficient storage – The user has exceeded their quota. Check at /user/id/quota for details. Mail the system administrator if you need more space on the server. By default, all users are allowed for 2000 models, 2000 BibTeX entries and 5 tasks running in parallel. The problem can be resolved if old models are deleted. Note that after a deletion, the models will be still on the server for 30 days (but no guarantee is provided) ; read http://opentox.ntua.gr/index.php/blog/69-oops-i-deleted-my-favorite-model for details.

Implementation Information	
HTTP Method	POST
Programming Language	The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://opentox.ntua.gr:8080 runs within an Apache Tomcat container.
Libraries used	WEKA version 3.6.0. was used for the implementation of this particular algorithm (see http://goo.gl/GMVkM). For a list of all dependencies of the JAQPOT3 project (https://github.com/alphaville/jaqpot3) read http://goo.gl/zNRjw .

Examples	
Example	<code>curl -X POST -d "dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545" -H subjectid:YOUR-TOKEN http://opentox.ntua.gr:8080/algorithm/mvh</code>

5.7 Domain of Applicability Estimation Algorithms

5.7.1 Leverage

Leverage is a well-known approach to the domain of applicability (DoA) estimation problem. The domain of applicability of a model defines whether the underlying model can be used to acquire a predicted value for a given compound. Leverage was implemented as a web service according to the standards of the version 1.2. for the OpenTox API. A deployed instance of the web service can be found online at <http://opentox.ntua.gr:8080/algorithm/leverages>.

General Information about the service	
Service description	The well-known leverages algorithm for the estimation of a model's applicability domain
URI	

http://opentox.ntua.gr:8080/algorithm/leverages (AA)
OpenTox API Reference www.opentox.org/dev/apis/api-1.2/Algorithm
Algorithm type Eager Learning, Single Target, Applicability Domain
Partner responsible for the implementation National Technical University of Athens, School of Chemical Engineering, Automatic Control Unit.
Contact within OT Pantelis Sopasakis < chvng@mail.ntua.gr >

Request/Response Information
Posted Parameters <p>dataset_uri (mandatory)</p> <p>As every model training algorithm, the data set URI is a mandatory parameter that has to be specified by the client. Unless it is a valid data set URI, a status code 400 is returned. (e.g. http://apps.ideaconsult.net:8080/ambit2/dataset/R545).</p>
Response <p>A task, which, upon successful completion, will be pointing to the URI of the trained model, is created and returned to the client.</p>

Status Codes	
202	Accepted - The request has succeeded and a task is returned to the client.
400	Bad Request - Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. Check under /dataset/id/feature for a list of features available. A 400 will also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled).
401	Unauthorized - The user is not authorized to perform the underlying operation or the user did not provide a valid token ID. It has been made easy to acquire a token using our online form at http://opentox.ntua.gr:8080/login - provide your credentials of (in case you don't have an account use the username <i>guest</i> and the same password).
404	The resource was not found - check your spelling.
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.
502	Bad Gateway - The service was unsuccessful because while the server was acting as a client, it received an unsuccessful response. In such a case, it seems that some other server is down. Troubleshooting is facilitated as in such a case an <i>Error Report</i> is returned to the client.
503	The service is not available for the time being - Try again later!
507	Insufficient storage - The user has exceeded their quota. Check at /user/id/quota for details. Mail the system administrator if you need more space on the server. By default, all users are allowed for

2000 models, 2000 BibTeX entries (see <http://opentox.ntua.gr:8080/bibtex>) and 5 tasks running in parallel. The problem can be resolved if you delete some of the model you don't need. Note that even if you do so, your models will be still on the server for 30days (but no guarantee is provided) ; read <http://opentox.ntua.gr/index.php/blog/69-ooops-i-deleted-my-favorite-model> for details.

Implementation Information

HTTP Method

POST

Programming Language

The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at <http://opentox.ntua.gr:8080> runs within an Apache Tomcat container.

Libraries used

WEKA version 3.6.0. was used for the implementation of this particular algorithm (see <http://goo.gl/GMVkM>). For a list of all dependencies of the JAQPOT3 project (<https://github.com/alphaville/jaqpot3>) read <http://goo.gl/zNRjw>.

Examples

Example

```
curl -X POST -d dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545 -d
prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200 -H subjectid:YOUR-TOKEN
http://opentox.ntua.gr:8080/algorithm/leverages
```

5.7.2 Several descriptor-based and structure-based applicability domain algorithms

AMBIT implementation of applicability domain estimation algorithms, as described in section 3.10.

General Information about the service

Service description

Applicability domain algorithms, URI can be retrieved via

<http://apps.ideaconsult.net:8080/ambit2/algorithm?type=AppDomain>

URI

<http://apps.ideaconsult.net:8080/ambit2/algorithm/pcaRanges>
<http://apps.ideaconsult.net:8080/ambit2/algorithm/distanceEuclidean>
<http://apps.ideaconsult.net:8080/ambit2/algorithm/distanceCityBlock>
<http://apps.ideaconsult.net:8080/ambit2/algorithm/distanceMahalanobis>
<http://apps.ideaconsult.net:8080/ambit2/algorithm/nparamdensity>
<http://apps.ideaconsult.net:8080/ambit2/algorithm/leverage>
<http://apps.ideaconsult.net:8080/ambit2/algorithm/fptanimoto>
<http://apps.ideaconsult.net:8080/ambit2/algorithm/fpmismissingfragments>

OpenTox API Reference

www.opentox.org/dev/apis/api-1.2/Algorithm

Algorithm type Applicability Domain
Partner responsible for the implementation Ideaconsult Ltd.
Contact within OT Nina Jeliaskova <jeliaskova.nina@gmail.com>

Request/Response Information
Posted Parameters <p>dataset_uri (mandatory)</p> <p>The data set URI is a mandatory parameter that has to be specified by the client. When assessing a training set of a model, this is the training set of the model.</p> <p>dataset_service (optional, specifies the URI of the dataset service, where the result will be stored. By default the dataset service ,where the training dataset reside is used.)</p>
Response <p>A task, which, upon successful completion, will be pointing to the URI of the applicability domain assessment (which becomes available as an OpenTox model itself), is created and returned to the client. The new model URI is then used to verify if a query compound is within or outside of the applicability domain.</p>

Status Codes	
202	Accepted – The request has succeeded and a task is returned to the client.
400	Bad Request – Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. Check under /dataset/id/feature for a list of features available. A 400 will be also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled).
404	The resource was not found
500	Internal Server Error – The parameters you posted are acceptable but some internal error has occurred.

Implementation Information
HTTP Method POST
Programming Language <p>The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://apps.ideaconsult.net:8080/ambit2 runs within an Apache Tomcat container.</p>
Libraries used

Main dependencies include JAMA (for matrix operations) and the CDK (for manipulating chemical structures). Full dependencies list available via maven pom.xml configuration of AMBIT project and Maven repository

Examples

Example

```
curl -X POST -d dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545 -d -H subjectid:YOUR-TOKEN http://apps.ideaconsult.net:8080/ambit2/algorithm/fptanimoto
```

5.8 Miscellaneous Algorithms

5.8.1 Compound, Similarity and Substructure Search

Returns search results as OpenTox Dataset, in several supported formats, including the mandatory RDF.

General Information about the service

Service description

A set of services, allowing to retrieve chemical compounds by various identifiers , exact, substructure or similarity search

URI

Search a compound by identifier (CAS, EINECS, chemical name, SMILES, InChI). If SMILES or InChI is provided, an exact search is performed.

<http://apps.ideaconsult.net:8080/ambit2/query/compound/IDENTIFIER/all>

or

<http://apps.ideaconsult.net:8080/ambit2/query/compound/search/all?search=IDENTIFIER>

or

<http://apps.ideaconsult.net:8080/ambit2/query/compound/url/all?uri=COMPOUND-URI>

Substructure search

<http://apps.ideaconsult.net:8080/ambit2/query/smarts?search=SMARTS>

Similarity (Tanimoto distance) search

<http://apps.ideaconsult.net:8080/ambit2/query/similarity?search=SMILES&threshold=TANIMOTO-DISTANCE-THRESHOLD>

OpenTox API Reference

www.opentox.org/dev/apis/api-1.2/Algorithm

Algorithm type

Applicability Domain

Partner responsible for the implementation

Ideaconsult Ltd.

Contact within OT

Nina Jeliazkova <jeliazkova.nina@gmail.com>

Request/Response Information

<p>Posted Parameters</p> <p>dataset_uri (optional)</p> <p>HTTP Header parameter "Accept:" defines the format of the returned results.</p>
<p>Response</p> <p>The search results, in the requested format. The results of a identifier search include CAS, chemical names, EINECS, REACH registration date , SMILES, InChI. The search results depend on the database content, as described in the Deliverable D3.4 Final Database.</p>

Status Codes	
200	OK - The request has succeeded and the results in the requested format are returned to the client.
400	Bad Request - Invalid parametrization; common mistakes are that the data set URI was misspelled or the feature URI was not included in the data set. Check under /dataset/id/feature for a list of features available. A 400 will be also occur in case the prediction feature is not numeric (e.g. String-valued features cannot be handled).
404	The resource was not found
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.

Implementation Information
<p>HTTP Method</p> <p>GET</p>
<p>Programming Language</p> <p>The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://apps.ideaconsult.net:8080/ambit2 runs within an Apache Tomcat container.</p>
<p>Libraries used</p> <p>AMBIT2 packages. Full dependencies list available via maven pom.xml configuration of Ambit project and Maven repository</p>

Examples
<p>Example</p> <p>Search a compound by identifier (CAS, EINECS, chemical name, SMILES, InChI)</p> <p>curl http://apps.ideaconsult.net:8080/ambit2/query/compound/249-323-0/all</p> <p>Substructure search</p> <p>curl http://apps.ideaconsult.net:8080/ambit2/query/smarts?search=c1cccc1</p> <p>Similarity (Tanimoto distance) search</p> <p>curl http://apps.ideaconsult.net:8080/ambit2/query/similarity?search=c1cccc1&threshold=0.85</p>

5.8.2 Structure Diagram Generation, Integrated with Compound Service

Returns 2D depiction of a chemical structure.

General Information about the service	
Service description	If HTTP header "Accept:image/png" or "Accept:image/gif" is provided , returns the 2D structure diagram http://apps.ideaconsult.net:8080/ambit2/compound/{ID}
URI	http://apps.ideaconsult.net:8080/ambit2/compound/{ID}
OpenTox API Reference	www.opentox.org/dev/apis/api-1.2/Compound
Algorithm type	Visualisation
Partner responsible for the implementation	Ideaconsult Ltd.
Contact within OT	Nina Jeliaskova <jeliaskova.nina@gmail.com>

Request/Response Information	
Posted Parameters	HTTP Header parameter "Accept:" defines the format of the returned results.
Response	An image in the requested format.

Status Codes	
200	OK - The request has succeeded and the results in the requested format are returned to the client.
404	The resource was not found
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.

Implementation Information	
HTTP Method	GET
Programming Language	The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://apps.ideaconsult.net:8080/ambit2 runs within an Apache Tomcat container.

Libraries used

AMBIT2 packages. Full dependencies list available via maven pom.xml configuration of Ambit project and Maven repository

Examples
Example

curl -H "Accept:image/png" <http://apps.ideaconsult.net:8080/ambit2/compound/1>

or shortcuts, accessible via a web browser:

<http://apps.ideaconsult.net:8080/ambit2/compound/1?media=image/png>

<http://apps.ideaconsult.net:8080/ambit2/compound/1/image>

5.8.3 Structure Diagram Generation by SMILES

Returns the 2D structure diagram of the compound, provided as SMARTS or InChI

General Information about the service
Service description

A set of services, allowing to retrieve 2D structure diagram by different providers

URI

<http://apps.ideaconsult.net:8080/ambit2/depict/cdk?search=c1cccc1>

<http://apps.ideaconsult.net:8080/ambit2/depict/daylight?search=c1cccc1>

<http://apps.ideaconsult.net:8080/ambit2/depict/cactvs?search=c1cccc1>

an HTML overview of the results by the three providers

<http://apps.ideaconsult.net:8080/ambit2/depict?search=c1cccc1>

OpenTox API Reference

www.opentox.org/dev/apis/api-1.2/Algorithm

Algorithm type

Visualisation

Partner responsible for the implementation

Ideaconsult Ltd.

Contact within OT

Nina Jeliaskova <jeliaskova.nina@gmail.com>

Request/Response Information
Posted Parameters

search (mandatory) – the compound to be visualized, in URL encoded SMILES or InChI

smarts (optional) – the substructure (specified by URL encoded SMARTS) to be highlighted

HTTP Header parameter "Accept:" defines the format of the returned results.

Response

An image, representing the structure diagram

Status Codes	
200	OK - The request has succeeded and the results in the requested format are returned to the client.
400	Bad Request - Invalid parametrization; Incorrect SMILES or SMARTS
404	The resource was not found
500	Internal Server Error - The parameters you posted are acceptable but some internal error has occurred.

Implementation Information
HTTP Method GET
Programming Language The project was built in Java and can run either as a standalone application or within a servlet container. The deployed instance at http://apps.ideaconsult.net:8080/ambit2 runs within an Apache Tomcat container.
Libraries used AMBIT2 packages. Full dependencies list available via maven pom.xml configuration of Ambit project and Maven repository

Examples
Example <pre>curl http://apps.ideaconsult.net:8080/ambit2/ambit2/depict/cdk?search=c1cccc1O&smarts=a</pre>

6 Conclusion

This document summarizes the work that has been accomplished within the OpenTox Framework regarding the development of the final (Q)SAR algorithms. A key decision towards this implementation was the adoption of the REST architectural style, because it is suitable for achieving three important goals: independent deployment of components, ease of standardized communication between components and generality of interfaces. The main contribution in the last year was the integration of in-house algorithms in the OpenTox framework and the application of these algorithms for modelling REACH-relevant endpoints. Another contribution was the development of OpenTox workflows for Taverna, which provides a graphical user interface to the OpenTox algorithms.

7 References

- [BUC10] Buchwald, F, Girschick, T, Frank, E, and Kramer, S (2010). *Fast Conditional Density Estimation for Quantitative Structure-Activity Relationships*. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, pp. 1268-1273, AAAI Press.
- [SOM07] Sommer, S and Kramer, S (2007). *Three Data Mining Techniques To Improve Lazy Structure-Activity Relationships for Non-Congeneric Compounds*. J. Chem. Inf. Model., **47**(6):2035-2043.

- [BUC11] Buchwald, F, Girschick, T, Seeland, M, and Kramer, S (2011). *Using Local Models to Improve (Q)SAR Predictivity*. *Molecular Informatics*, **30**(2–3):205–218.
- [SEE10] Seeland, M, Girschick, T, Buchwald, F, and Kramer, S (2010). *Online Structural Graph Clustering using Frequent Subgraph Mining*. In: *Proceedings of the European Conference of Machine Learning 2010*, ed. by J.L. Balcazar, F. Bonchi, A.Gionis, M.Sebag, vol. 3, pp. 213–228.
- [MAU09] Maunz, A, Helma, C, and Kramer, S (2009). *Large-scale Graph Mining Using Backbone Refinement Classes*. In: *KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 617–626, New York, NY, USA, 2009. ACM.
- [MAU10] Maunz, A, Helma, C, Cramer, T, and Kramer, S (2010). *Latent Structure Pattern Mining*. In *Proceedings of ECML PKDD 2010*, pp. 353–368. Springer Berlin / Heidelberg.
- [MAU11] Maunz, A, Helma, C, and Kramer, S (2011). *Efficient Mining for Structurally Diverse Subgraph Patterns in Large Molecular Databases*. *Mach. Learn.*, **83**:193–218.
- [NET2005] Netzeva, TI, Worth, AP, Aldenberg, T, Benigni, R, Cronin, MDT, Gramatica, P, Jaworska, JS, Kahn, S, Klopman, G, Marchant, CA, Myatt, G, Nikolova–Jeliazkova, N, Patlewicz, GY, Perkins, Y, Roberts, DW, Schultz, TW, Stanton, DT, van de Sandt, JJM, Tong, W, Veith G, and Yang, C (2005), *ECVAM WORKSHOP REPORT [Current status of methods for defining the applicability domain of \(quantitative\) structure–activity relationships](#)*, *ATLA*, **33**:155–173
- [JAW2007] Jaworska, J, Nikolova–Jeliazkova, N (2007), [How can structural similarity analysis help in category formation](#), *SAR QSAR Environ. Res.*, **18**:3–4
- [JAW2005] Jaworska, J, Nikolova–Jeliazkova, N, and Aldenberg, T, (2005) [QSAR Applicability Domain Estimation by Projection of the Training Set in Descriptor Space: A Review](#), *ATLA*, **33**:445–459
- [RYD2010] Rydberg, P, Gloriam, DE, Zaretski, J, Breneman, C, and Olsen, L, (2010), *SMARTCyp: A 2D Method for Prediction of Cytochrome P450–Mediated Drug Metabolism*, *ACS Med. Chem. Lett.*, **1**(3):96–100
- [LEE2008] Lee, AC, Yu, JY, and Crippen, GM, (2008), *pKa Prediction of Monoprotic Small Molecules the SMARTS Way*, *J. Chem. Inf. Model.*, **48**(10):2042–2053
- [ZHE2009] Zheng, M, Luo, X, Shen, Q, Wang, Y, Du, Y, Zhu, W, and Jiang, H (2009), *Site of metabolism prediction for six biotransformations mediated by cytochromes P450*. *Bioinformatics*, **25**(10):1251–1258
- [JEL2010] Jeliazkova, N, Zhiryakova, D, Aleksiev, B, and Jeliazkov, V (2010), *Comparison of CYP450–Mediated Metabolism Prediction Models*, *Application of the OpenTox Framework for Predictive Toxicology, QSAR2010*, poster
- [LOW2011] Lowe, DM, Corbett, PT, Murray–Rust, P, and Glen, RC (2011), *Chemical Name to Structure: OPSIN, an Open Source Solution*, *J. Chem. Inf. Model.*, **51**(3):739–753
- [SAR2003] Sarimveis, H, Alexandridis, A and Bafas, G (2003), *A fast algorithm for RBF networks based on subtractive clustering*, *Neurocomputing*, **51**:501–505
- [HAN1995] Hansch, C, Leo, A and Hoekman, D (1995), *Exploring QSAR, hydrophobic, electronic and steric constants*, ACS, Washington DC
- [TOD2000] Todeschini R, Consonni, V, Mannhold, R, Kubinyi, H and Timmerman, H (2000), *Handbook of Molecular descriptors*
- [CRA78] Cramer GM, Ford, RA and Hall, RL (1978), *Estimation of Toxic Hazard – Decision Tree Approach*, *J. Cosmet. Toxicol.*, **16**:255 –276, Pergamon Press

- [VER92] Verhaar HJM, van Leeuwen CJ and Hermens JLM (1992), *Classifying environmental pollutants. Structure-activity relationships for prediction of aquatic toxicity*. Chemosphere **25**:471– 491
- [WAL05] Walker, JD, Gerner, I, Hulzebos, E and Schlegel K (2005), *The Skin Irritation Corrosion Rules Estimation Tool (SICRET)*, QSAR Comb. Sci., **24**:378–384
- [GER05] Gerner, I, Liebsch, M and Spielmann, H (2005), *Assessment of the eye irritating properties of chemicals by applying alternatives to the Draize rabbit eye test: the use of QSARs and in vitro tests for the classification of eye irritation*, Alternatives to Laboratory Animals, **33**:215–237
- [BEN07] Benigni, R, Bossa, C, Netzeva, T, Rodomonte, A and Tsakovska, I (2007) *Mechanistic QSAR of aromatic amines: new models for discriminating between mutagens and nonmutagens, and validation of models for carcinogens*. Environ. Mol. Mutagen., **48**:754–771
- [BEN08] Benigni, R, Bossa, C, Jeliazkova, N, Netzeva, N, and Worth, A (2008), *The Benigni / Bossa rulebase for mutagenicity and carcinogenicity - a module of Toxtree*, JRC Scientific and Technical Reports, ecb.jrc.it/documents/QSAR/EUR_23241_EN.pdf

8 Appendix A

8.1 A.1 gSpan output

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ot="http://www.opentox.org/api/1.1#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:bo="http://www.blueobelisk.org/ontologies/chemoinformatics-#/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ota="http://www.opentox.org/algorithms.owl#">
  <owl:Class rdf:about="http://www.opentox.org/api/1.1#Algorithm"/>
  <owl:Class rdf:about="http://www.opentox.org/api/1.1#Parameter"/>
  <ot:algorithm rdf:about="http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/gSpan">
    <owl:sameAs>http://www.blueobelisk.org/ontologies/chemoinformatics-
    algorithms/#subgraph</owl:sameAs>
    <ot:parameters>
      <ot:Parameter>
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >embeddingLists</dc:title>
        <dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Use embedding lists (Default: 0).</dc:description>
        <ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >optional</ot:paramScope>
        <ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
          >>false</ot:paramValue>
      </ot:Parameter>
    </ot:parameters>
    <dc:contributor>tobias.girschick@in.tum.de</dc:contributor>
    <ot:parameters>
      <ot:Parameter>
        <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >symmetries</dc:title>
        <dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```

  >Use symmetries (Default: 0).</dc:description>
  <ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >optional</ot:paramScope>
  <ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >>false</ot:paramValue>
</ot:Parameter>
</ot:parameters>
<ot:parameters>
  <ot:Parameter>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >dataset_uri</dc:title>
    <dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >URI to the dataset to be used.</dc:description>
    <ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >mandatory</ot:paramScope>
    <ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    ></ot:paramValue>
  </ot:Parameter>
</ot:parameters>
<dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>OpenTox REST interface to the qSpan' algorithm implementation of TUM.</dc:description>
<dc:creator>fabian.buchwald@in.tum.de</dc:creator>
<rdf:type>http://www.opentox.org/algorithms.owl#Unsupervised</rdf:type>
<rdf:type rdf:resource="http://www.opentox.org/api/1.1#Algorithm"/>
<rdf:type>http://www.opentox.org/algorithms.owl#DescriptorCalculation</rdf:type>
<dc:contributor>joerg.wicker@in.tum.de</dc:contributor>
<dc:identifier rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI"
>http://opentox.informatik.tu-muenchen.de:8080/OpenTox/algorithm/qSpan</dc:identifier>
<ot:parameters>
  <ot:Parameter>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >dataset_service</dc:title>
    <dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >URI to the dataset service to be used (Default:
    http://apps.ideaconsult.net:8080/ambit2/dataset).</dc:description>

```

```

<ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>optional</ot:paramScope>
<ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
></ot:paramValue>
</ot:Parameter>
</ot:parameters>
<ot:parameters>
<ot:Parameter>
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>linearFragments</dc:title>
<dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Restrict search to linear fragments (Default: 0).</dc:description>
<ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>optional</ot:paramScope>
<ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</ot:paramValue>
</ot:Parameter>
</ot:parameters>
<rdf:type>http://www.opentox.org/algorithms.owl#PatternMining</rdf:type>
<ot:parameters>
<ot:Parameter>
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>minSup</dc:title>
<dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Specifies the min support for mining (absolute) (Default: 20).</dc:description>
<ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>optional</ot:paramScope>
<ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>10</ot:paramValue>
</ot:Parameter>
</ot:parameters>
<ot:parameters>
<ot:Parameter>
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>fragmentsWithMaxEdges</dc:title>

```

```

<dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Restrict search to fragments with maximum i edges. (Default: 0)</dc:description>
<ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>optional</ot:paramScope>
<ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</ot:paramValue>
</ot:Parameter>
</ot:parameters>
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>qSpan'</dc:title>
<ot:parameters>
<ot:Parameter>
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>acyclicFragments</dc:title>
<dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Restrict search to acyclic fragments (Default: 0).</dc:description>
<ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>optional</ot:paramScope>
<ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
>>false</ot:paramValue>
</ot:Parameter>
</ot:parameters>
<dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>Fri Aug 26 12:52:39 CEST 2011</dc:date>
<ot:parameters>
<ot:Parameter>
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>NumMaxEdges</dc:title>
<dc:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Number of maximum i edges. (Feature corresponds to fragmentsWithMaxEdges, it can only be set if
fragmentsWithMaxEdges is set to 1) (Default: 5).</dc:description>
<ot:paramScope rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>optional</ot:paramScope>
<ot:paramValue rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>5</ot:paramValue>

```

```
</ot:Parameter>
</ot:parameters>
</ot:algorithm>
<owl:AnnotationProperty rdf:about="http://purl.org/dc/elements/1.1/title"/>
</rdf:RDF>
```

8.2 A.2. Learn and validate a LoMoGraph model

Learn a LoMoGraph model

```
curl -i -X POST -d 'dataset_service=http://apps.ideaconsult.net:8080/ambit2/dataset' -d
'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545' -d
'prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200' -d 'weka_parameter_string=-W
weka.classifiers.functions.LinearRegression -S 1 -C -R 1.0E-8' -d 'minimumClusterSize=20' -d 'threshold=0.4'
-H 'subjectid:someToken' http://opentox-dev.informatik.tu-muenchen.de:8080/OpenTox-
dev/algorithm/LoMoGraphRegression
```

Validate the LoMoGraph model from above via 10 fold cross validation:

```
curl -X POST -d algorithm_uri="http://opentox-dev.informatik.tu-muenchen.de:8080/OpenTox-
dev/algorithm/LoMoGraphRegression" -d
'dataset_uri=http://apps.ideaconsult.net:8080/ambit2/dataset/R545' -d
'prediction_feature=http://apps.ideaconsult.net:8080/ambit2/feature/22200' -d num_folds=10 -d
algorithm_params="weka_parameter_string=-W weka.classifiers.functions.LinearRegression -S 1 -C -R 1.0E-
8%3BminimumClusterSize=20%3Boptthreshold=0.4" http://opentox.informatik.uni-
freiburg.de/validation/crossvalidation -H 'subjectid:AQIC5wM2LY4SfcyCbpxgKyCjemMQZCwD-
kITimvgk4jUYj0.*AAJTSQACMDE.*'
```