Use lazar-GUI for predictions with QSAR models based on the lazar-OpenTox framework

David Vorgrimmler and Micha Rautenberg, *in silico* toxicology Basel, Switzerland

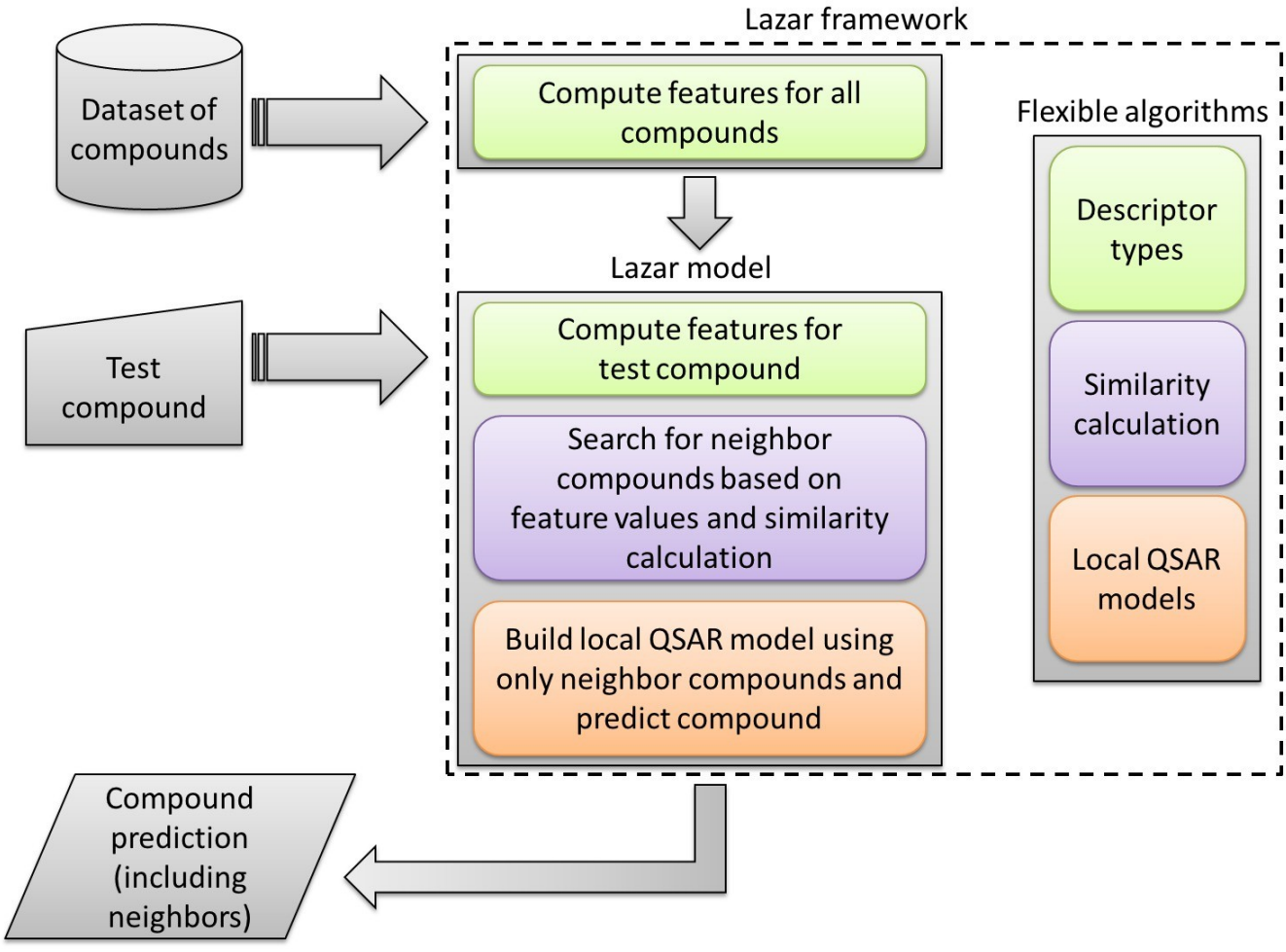
Introduction

lazar[1] (lazy structure-activity relationships) is a modular framework for predictive toxicology. Similar to the read across procedure in toxicological risk assessment, *lazar* creates local QSAR (quantitative structure-activity relationship) models for each compound to be predicted. Model developers can choose between a large variety of algorithms for descriptor calculation and selection, chemical similarity indices for model building. The *lazar*-GUI is a graphical webinterface for the *lazar* framework, which enables user to make predictions on existing QSAR models.
In this workshop participants will use the *lazar*-GUI to make predictions for chemical compounds on classification and regression models. The available models are based on the *lazar*-OpenTox framework.

Full compatibility with OpenTox

Each component and algorithm of the *lazar* framework is implemented as a OpenTox[2] web service . This makes our framework modular, each algorithm can be easily replaced. Each web service can be used standalone (e.g. algorithm, dataset, model, ...). The source code of the framework is available as open-source on GitHub (https://github.com/opentox). The *lazar*-GUI is located at https://services.in-silico.ch/predict.

Workflow



Descriptor types

Substructure fragments :

- Substructure descriptors are binary descriptors that do occur in a compound or not.
- *lazar* integrates algorithms to mine endpoint relevant substructures.
  - Backbone Refinement Class Mining (BBRC)
  - Latent Structure Pattern Mining (LAST)

Physico-chemical (PC) properties

- Numerical descriptors (e.g. molecular weight, largest chain, logP, surface charge).
- *lazar* utilizes open source cheminformatics libraries (Chemistry Development Kit (CDK), OpenBabel, JOELib) to compute PC descriptors.

Also Measured properties (experimental measurements) can be utilized by *lazar*.

Similarity calculation

Substructures

- A weighted Tanimoto index searches for similar compounds (neighbors) to the query compound in the training dataset .
- Tanimoto similarity encodes presence or absence of substructures in molecules
- Optionally, the number of times a substructure occurs in each compound is taken into account

Physico-chemical properties / numerical features

- Cosine similarity, based on the PCA-transformed feature space, is used to find the neighbors
- Cosine similarity computes the distance between two compounds by measuring the angle between the feature value vectors

Local QSAR models

Only neighbor compounds are used to build a local QSAR model.

Available learning algorithms

- Classification: weighted majority voting and support vector machines (SVM)
- Regression: support vector machines (SVM) with a variety of kernels

Applicability Domain

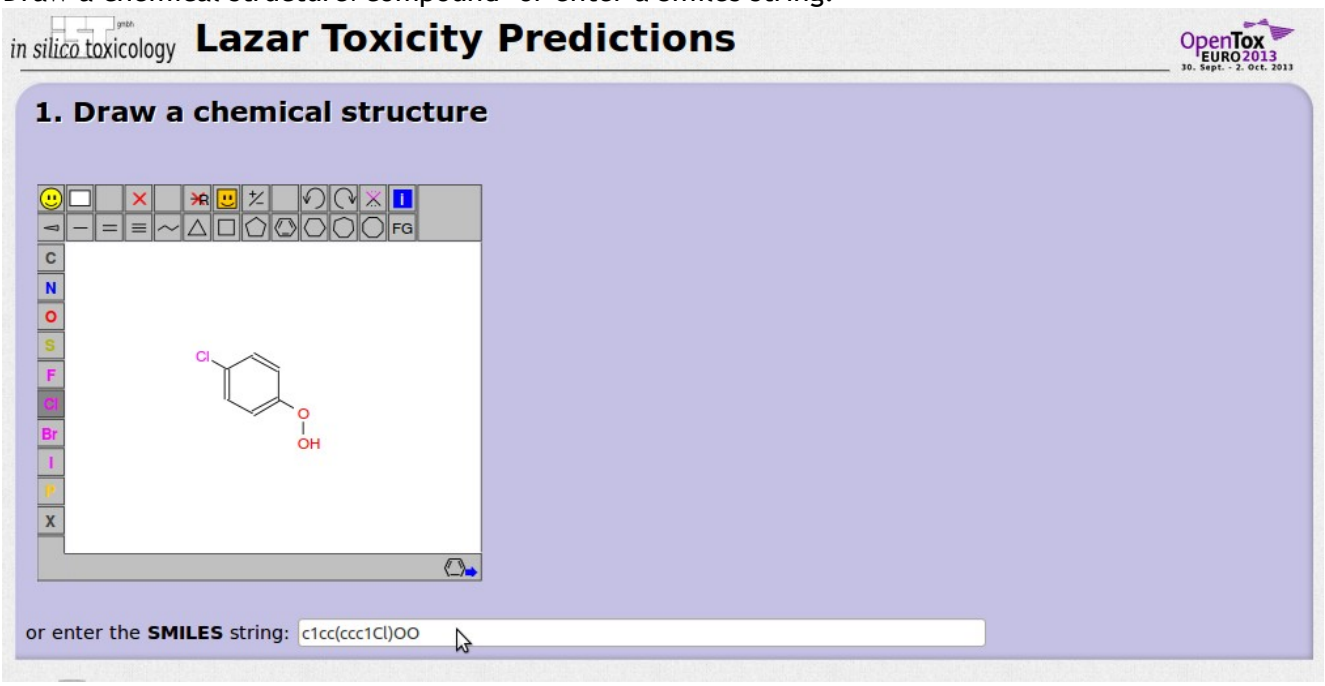
- *lazar* provides a confidence value with each prediction, that measures how
- 'certain' the algorithm is about the prediction
- The confidence is based on:
  - The number of compounds (neighbors) that are similar to the query structure
  - The degree of similarity of the neighbors to the query compound
  - The coherence of experimental data within neighbors.

Hands-on Workshop Session

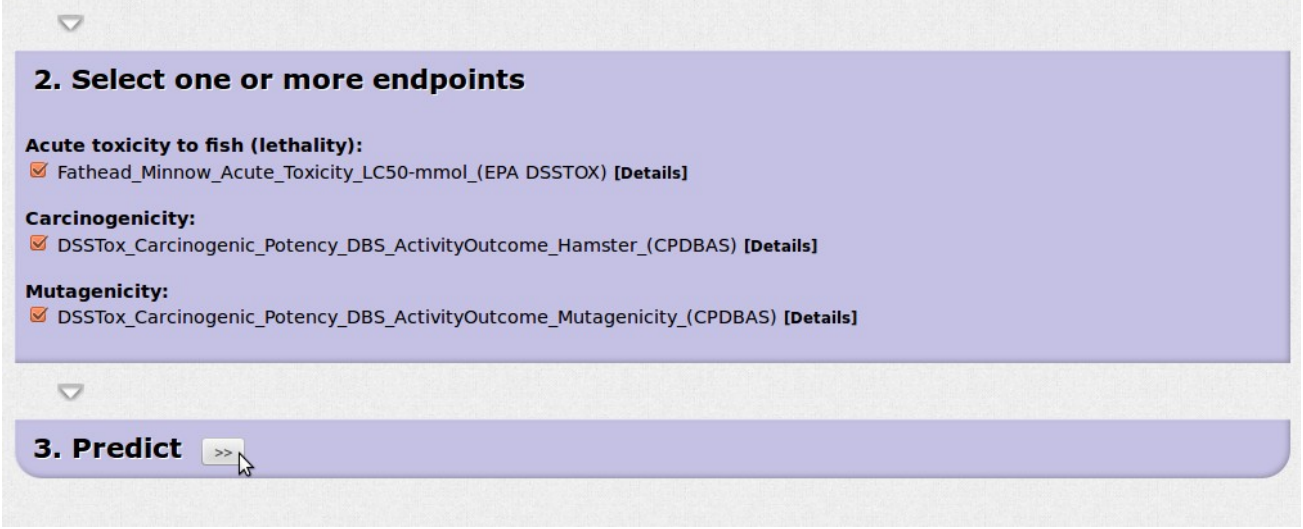
Requirements: Browser with Javascript activated (we recommend Firefox), Java installed, Internet connection.

Part 1 - Make a prediction

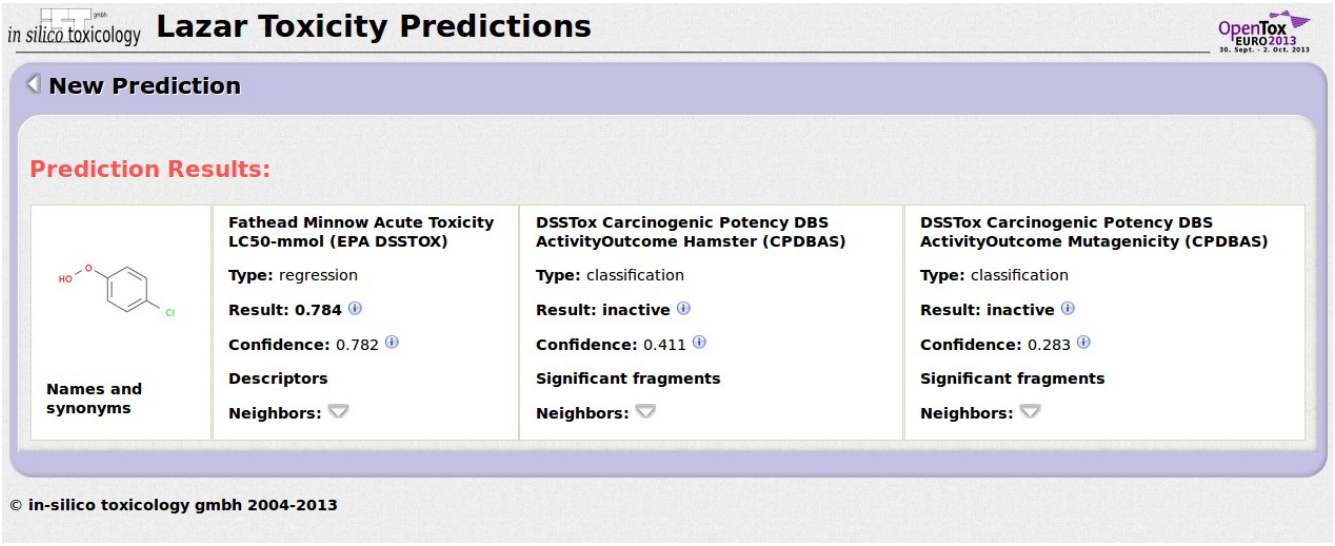
Draw a chemical structure/compound[3] or enter a Smiles string.



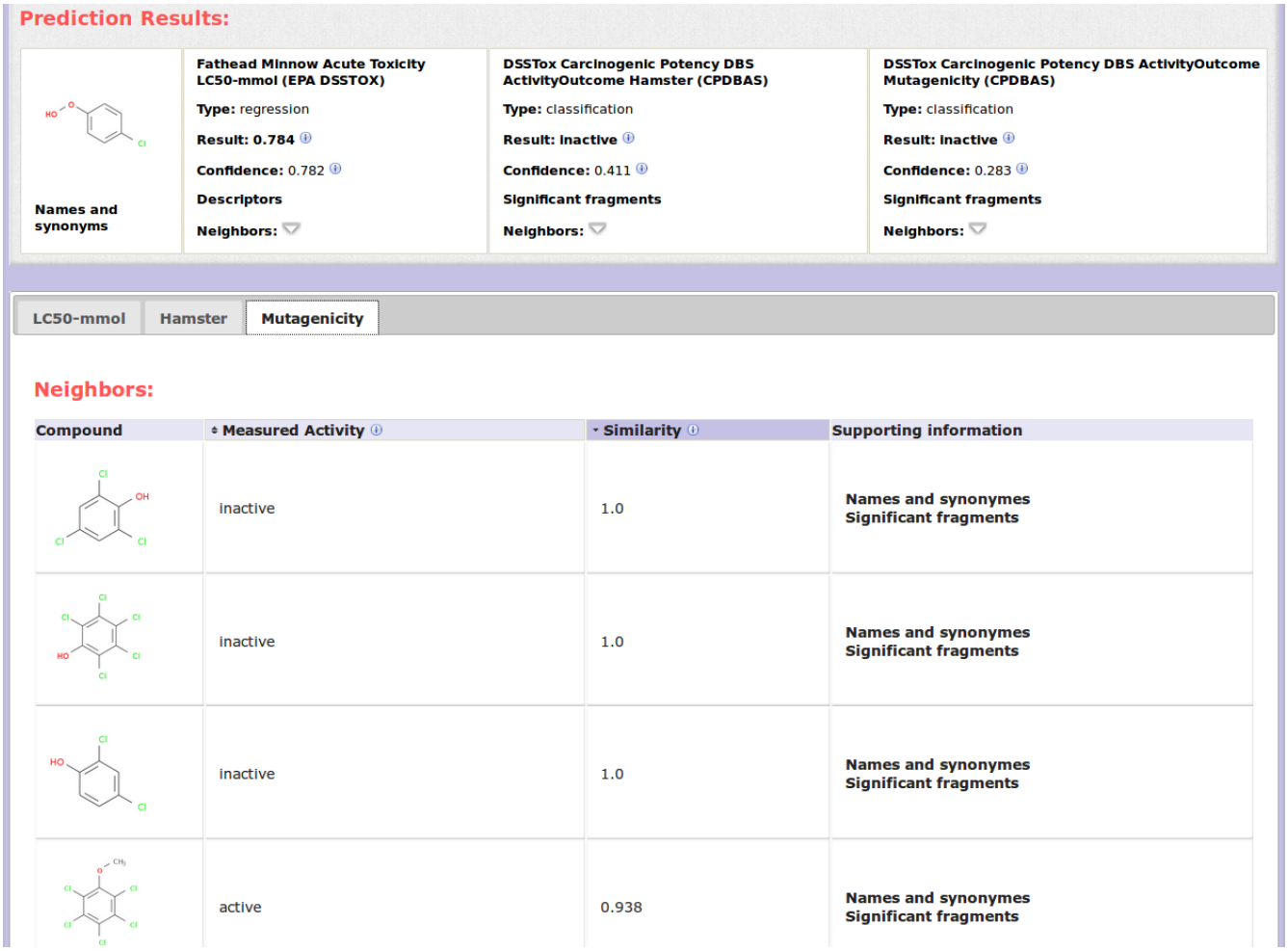
Selected one or more endpoints. You find additional informations and a link to validation-results when you click on **[Details]**



Run a prediction, click on “Predict” and wait for the result page.

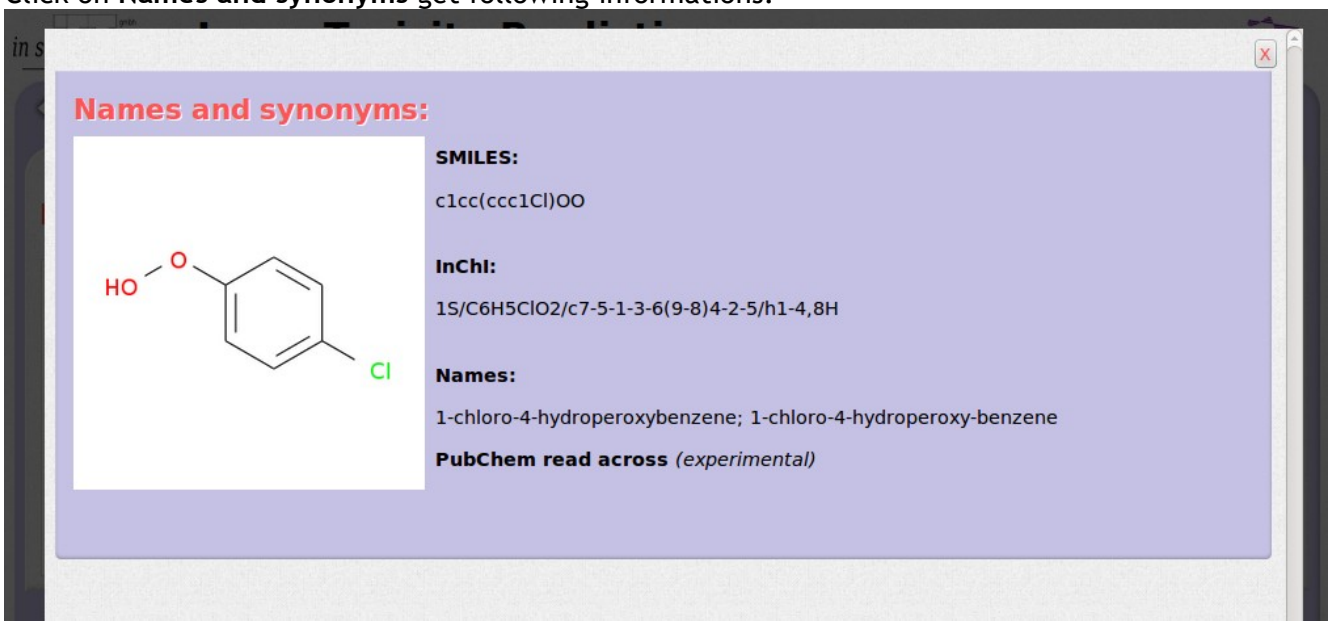


Click on **Neighbors** to examine the predicted compound’s neighbors with relevant substructures, measured activity and similarity.



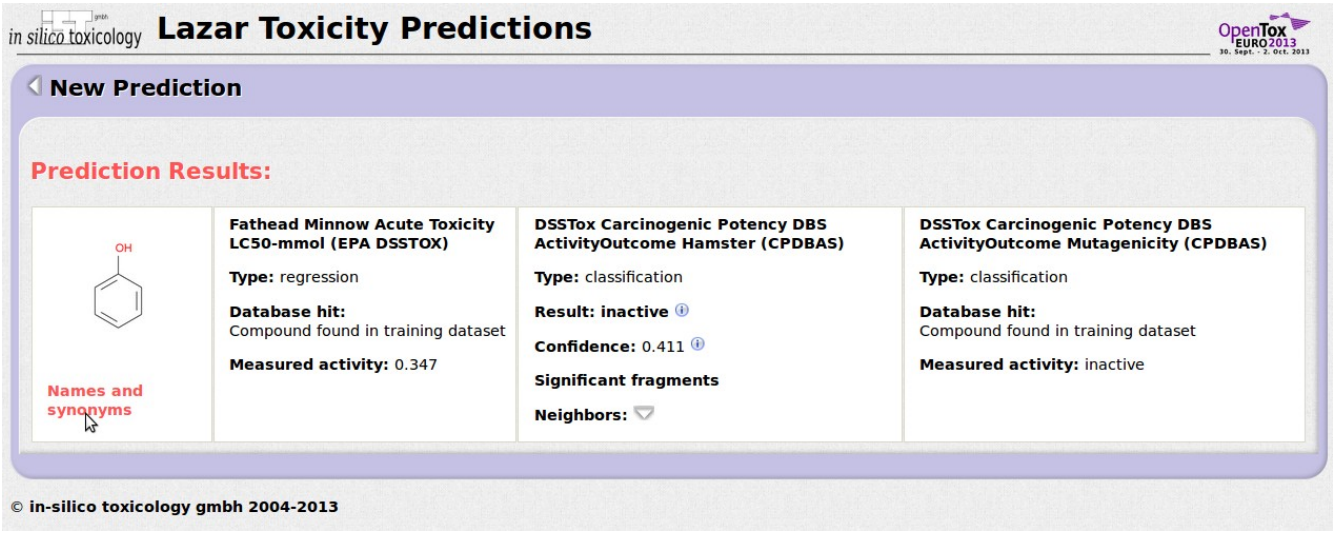
Click on **Descriptors** for regression or **Significant fragments** for classification models to inspect the given features of the compound.
Descriptors shows the description of the physico-chemical properties and its values.
Significant fragments shows structural features sorted by their effects (see vocabulary). Each effect list is sorted by the p value of the substructure which was calculated during the feature generation. The substructures are written in SMARTS notation[4].

Click on **Names and synonyms** get following informations.



Part 2 - Make a prediction and a PubChem read across investigation

Click on **New Prediction**
Draw a chemical structure or enter a Smiles string e.g. “c1ccc(cc1)O” (PHENOL).
Selected one or more endpoints and start a prediction.



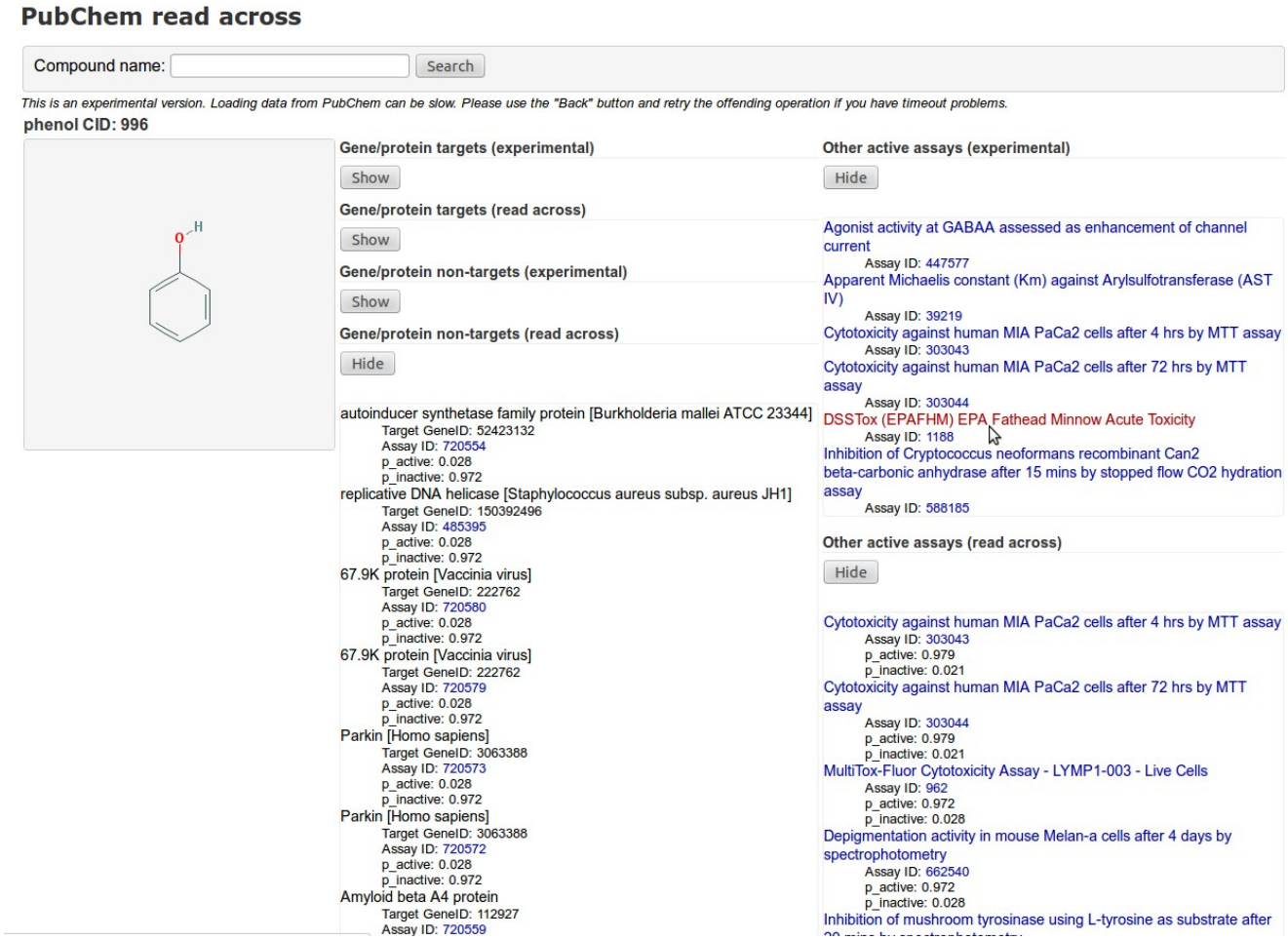
There are two database hits for this compound.

Click on **Names and synonyms** and copy “PHENOL” for the PubChem read across search.

Click on **PubChem read across** to open a new tab with our service. Enter and search for “PHENOL”.

In your investigation you will find a lot of information for this compound including the underlaying assays of our models.

Please keep in mind that our PubChem read across service is still experimental.



Vocabulary

- **Classification** - Prediction of qualitative properties, e.g. to distinguish between toxic and non-toxic compounds
- **Confidence** - Indicates the applicability domain of a model. Predictions with a high confidence can be expected to be more reliable than predictions with low confidence. Confidence values may take any value between 0 and 1. For most models confidence > 0.025 is a sensible (hard) cutoff to distinguish between reliable and unreliable predictions.
- **InChI** - International Chemical Identifier
- **Measured Activity** - Experimental result(s) from the training dataset.
- **Regression** - Prediction of quantitative properties, e.g. LC50 values
- **Result** - *lazar* calculates searches the training dataset for similar compounds (neighbors) and calculates the prediction from their measured activities. *lazar* calculates predictions using
    - a majority vote (weighted by compound similarity) for classification (original publication)
    - a local QSAR model based on neighbors for regression (original publication)

  Please keep in mind that predictions are based on the measured activities of neighbors.
- **Significant fragments** - Substructures that occur (statistically significant) more frequently in one of the toxicity classes or correlate with quantitative toxicity values. Substructures can take any shape (without cycles) and are determined with the fminer algorithm.
- **Similarity** - *lazar* calculates activity specific similarities based on the presence of statistically significant fragments.
  This procedure will
    - consider only those parts of a chemical structure that are relevant for a particular endpoint
    - ignore inert parts of the structure
    - lead to different similarities, depending on the toxic endpoint.
      Similarities of 1 may be encountered even for structurally dissimilar compounds, because inert parts are ignored.
- **SMILES** - Simplified molecular-input line-entry system

Literature

➢ 1. Maunz A., Gütlein M., Rautenberg M., Vorgrimmler D., Gebele D., Helma C.: Lazar: A Modular Predictive Toxicology Framework. Frontiers in Pharmacology 4(38), April 2013. http://www.frontiersin.org/predictive_toxicity/10.3389/fphar.2013.00038/abstract
➢ 2. Hardy, B., Douglas, N., Helma, C., Rautenberg, M., Jeliazkova, N., et al. (2010). Collaborative development of predictive toxicology applications. J. Cheminform. 2, 7.
➢ 3. JSME Molecule Editor by Peter Ertl and Bruno Bienfait http://peter-ertl.com/jsme/
➢ 4. SMARTS - A Language for Describing Molecular Patterns http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html

About us

in silico toxicology is a small research oriented company focusing on the development of open source predictive toxicology software and services. We are specialized in developing and applying advanced data mining algorithms for complex chemical and biological data.

Contact

in silico toxicology gmbh
Burgfelder Str. 190
CH-4055 Basel, Switzerland

Email: support@in-silico.ch