

Knowledge Fair Table G - Validate your QSAR Model and Create a Report for Developers Rodos Palace Hotel, Rhodes, Greece, 19th September 2010

Objective

Learn to handle the validation and reporting services for models using curl calls.

Summary

In this knowledge fair, we will demonstrate how to effectively use the validation and reporting web services behind the applications of ToxPredict and ToxCreate. Using curl calls, we will contact the web services and validate a model or algorithm using a number of different approaches such as k-fold split, training-test-split, or bootstrapping. Furthermore, we will generate a QMRF report and visualize it using the QMRF Editor web start application. This knowledge fair table is aimed at developers, who want to look behind the scenes.

API-Definition

Before we start, it is probably an advantage to have a window with the API definitions for the validation services open in a browser: so please open the following link in a browser, preferably Firefox:

<http://www.opentox.org/data/documents/development/validation/Validation/>

Validation Examples

In this part we will have a look how to access the validation web services using the command line tool curl (<http://curl.haxx.se>).

First we want to list all available validations. To do that, it is necessary, to execute the following command in a terminal window:

```
curl http://opentox.informatik.uni-freiburg.de/validation
```

Validate an algorithm on a dataset via training-test-split

This will create a new validation object. A model is constructed by splitting a dataset into two parts: one for learning a model and one for testing, i.e. predicting and estimating the performance of the constructed model. Splitting the dataset is done in random fashion. One can also define the ratio for splitting into training and test, the default is 67% training and 33% test.

```
curl -X POST -d algorithm_uri="http://opentox.informatik.uni-freiburg.de/algorithm/lazar" -d  
dataset_uri="http://opentox.informatik.uni-freiburg.de/dataset/1" -d  
prediction_feature="http://localhost/toxmodel/feature%23Hamster%2520Carcinogenicity%2520(DSSTOX/  
CPDB)" -d algorithm_params="feature_generation_uri=http://opentox.informatik.uni-  
freiburg.de/algorithm/fminer" -d split_ratio=0.9 -d random_seed=2 http://opentox.informatik.uni-  
freiburg.de/validation/training_test_split
```

Validating algorithms may be a time consuming task. Therefore the result of the above curl call is a task URI. To query the status of the task URI, enter the following command in the terminal (where the term <TASK-ID> should be replaced with the correct task ID).

```
curl http://opentox.informatik.uni-freiburg.de/task/<TASK-ID>
```

As soon as the task is completed, your validation is available. The validation URI can be found in the resultURI property of the task:

```
---
:uri: http://opentox.informatik.uni-freiburg.de/task/<id>
:hasStatus: Completed
:resultURI: http://opentox.informatik.uni-freiburg.de/validation/<VALIDATION-ID>
[...]
```

Use curl to get a closer look at your validation result:

```
curl http://opentox.informatik.uni-freiburg.de/validation/<VALIDATION-ID>
```

Just like the task result, the validation result is formatted in YAML, a markup language that is human readable. Have a look at the statistics like area-under-roc, or confusion matrix values.

Validate an algorithm on a dataset via bootstrapping

Bootstrapping is a machine learning technique that splits a dataset into training and test set via "sampling with replacement".

```
curl -X POST -d algorithm_uri="http://opentox.informatik.uni-freiburg.de/algorithm/lazar" -d
dataset_uri="http://opentox.informatik.uni-freiburg.de/dataset/1" -d
prediction_feature="http://localhost/toxmodel/feature%23Hamster%2520Carcinogenicity%2520(DSSTOX/
CPDB)" -d algorithm_params="feature_generation_uri=http://opentox.informatik.uni-
freiburg.de/algorithm/fminer" -d random_seed=2 http://opentox.informatik.uni-
freiburg.de/validation/bootstrapping
```

Again, this curl call returns a task. As soon as the bootstrapping validation is finished, you validation is provided as before.

Validation Reports

Validation reports present validation results in a nice human readable format. This curl call gives you a list of available validation reports:

```
curl http://opentox.informatik.uni-freiburg.de/validation/report/validation
```

Create validation report from validation

This curl call will create a report for the validation that you just performed before. Choose which validation you like (training-test split or bootstrapping).

```
curl -X POST -d validation_uris="http://opentox.informatik.uni-freiburg.de/validation/<VALIDATION-
ID>" http://opentox.informatik.uni-freiburg.de/validation/report/validation
```

A report is created that is wrapped in a task URI as above.

You can access you report in YAML-format with the following curl call (this time you have to specify YAML as requested format manually, as the default report format is 'text/html')

```
curl -H "accept:application/x-yaml" http://opentox.informatik.uni-
freiburg.de/validation/report/validation/<REPORT-ID>
```

You can also view this report via a web browser, where connected information for this validation object is available. Use you open web browser and open a new tab and simply enter

<http://opentox.informatik.uni-freiburg.de/validation/report/validation/<REPORT-ID>>

in the address line of the browser.

Create a QMRF Report

QMRF (QSAR Model Reporting Format) is a harmonized template by the European Commission for summarizing and reporting key information on QSAR models.

A QMRF is created for a particular QSAR model. To this end, you can build a model on the complete dataset we were using the so far with the following curl call:

```
curl -d dataset_uri="http://opentox.informatik.uni-freiburg.de/dataset/1" -d
prediction_feature="http://localhost/toxmodel/feature%23Hamster%2520Carcinogenicity%2520(DSSTOX/
CPDB)" -d feature_generation_uri="http://opentox.informatik.uni-freiburg.de/algorithm/fminer"
http://opentox.informatik.uni-freiburg.de/algorithm/lazar
```

Use the new model to build a QMRF report via:

```
curl -X POST -d model_uri=http://opentox.informatik.uni-freiburg.de/model/<MODEL-ID>
http://opentox.informatik.uni-freiburg.de/validation/reach_report/QMRF
```

This report can be accessed via curl:

```
curl http://opentox.informatik.uni-freiburg.de/validation/reach_report/QMRF/<REPORT-ID>
```

Alternatively, use the QMRF editor to edit this report by visiting the address with your browser:

http://opentox.informatik.uni-freiburg.de/validation/reach_report/QMRF/<REPORT-ID>/editor

Further validation techniques

For a bit more technical description and further examples including:

- how to validate a model on a test dataset
- how to validate an algorithm on a training and test dataset
- how to create a validation object by comparing feature values
- how to validate an algorithm on a dataset via k-fold cross-validation
- and more

have a look at the examples web page located at:

<http://opentox.informatik.uni-freiburg.de/validation/examples>