# Case Study
Reverse dosimetry and PBPK prediction [RevK]

## SUMMARY

## DESCRIPTION
- Implementation team
- Case Study objective
- Risk assessment framework

## DEVELOPMENT
- Use Cases Associated
- Databases and tools
- Technical implementation

## OUTCOMES

## REFERENCES

## APPENDIX
- Integration of PKsim in OpenRiskNet Infrastructure through the Jaqpot modelling platform
- Integration of httk in OpenRiskNet Infrastructure through the Jaqpot modelling platform
- Integration of custom made PBPK models in the Jaqpot modelling platform
SUMMARY

This case-study demonstrates and document the use of a web interface to physiologically-based pharmacokinetic models for forward and reverse dosimetry calculations. Forward calculations compute internal concentrations from given exposure doses. Reverse calculations compute exposure doses from internal concentrations or measured biomarker levels (e.g., urine concentration data). The result of those calculations can be used in risk assessments to help with in vitro to in vivo extrapolations or interspecies extrapolations.

The tools used for this case-study will be developed at NTUA and accessible as user-friendly web services for simulations with the httk PBPK model (httk is a R package) or with the PKSim model, or with a user-defined model.

For each of these three models, forward and reverse simulations will be performed for a referenced chemical: Imazalil for the httk model, Theophylline for the PKSim model, and Diazepam for the user-defined model. The exposure scenarios chosen will be in the range of corresponding environmental or therapeutic levels.

The conduct of the simulations and their results will be documented in the format of a User’s Manual which will be accessible to future users.
DESCRIPTION

Implementation team
CS leader: INERIS; Other team members: NTUA
- The code and web implementation will be developed by NTUA
- The case-studies' simulations will be run by INERIS
- Code and technical service documentation will be provided by NTUA. INERIS will document the users’ operations and results of the case-study demonstrations.

Case Study objective
The objective of this case study is to demonstrate and document the capabilities of the OpenRiskNet-developed web-services for PBPK modeling. Both forward and reverse dosimetry predictions will be illustrated.

Risk assessment framework
The application frameworks are, for example: REACH risk assessments; SEVESO II directive on safety around industrial plants; Internal chemical, cosmetic, or pharmaceutical company assessments of workers’ safety, or consumer’s safety. All those require integration and extrapolation of in vitro and/or in vivo data on animals to assess human risks.
DEVELOPMENT

Use Cases Associated

In the future, web-services for QSAR predictions of partition coefficients for chemicals or other PBPK parameters could be used for input in the service’s PBPK models.

Databases and tools

We will use open source software able to implement PBPK models: R (with package httk), PKSim, GNU MCSim. The Jaqpot biokinetic services will be used to publish the PBPK models as web services. Service clients will be developed in Python or as Jupyter notebooks.

Databases of parameter values will be provided by the httk R package, and the PKSim model.

Technical implementation

**Implementation of the chosen PBPK model as web services:**

PBPK models for a specific class of chemicals and animal species will be selected by the user from a particular PBPK modelling environment (e.g., httk in R, PKSim, GNU MCSim).

The chosen PBPK model will be exposed as a web service using the Jaqpot modelling platform. This is possible through the Jaqpot Protocol of Data Interchange (JPDI) which allows to dynamically and seamlessly incorporate practically any algorithmic implementation into Jaqpot. The protocol specifies the form of data exchange between Jaqpot services and third party algorithm web service implementations. The Jaqpot framework already provides wrappers for the R language and the Python language. Integration with R is made possible through the OpenCPU system, which defines an HTTP API for embedded scientific computing based on R, although this approach could easily be generalized to other computational back-ends (Ooms, 2014). OpenCPU acts as a wrapper to R that is readily able to expose R functions as RESTful HTTP resources. The OpenCPU server takes advantage of multi-processing in the Apache2 web server to handle concurrency. This implementation uses forks of the R process to serve concurrent requests immediately with little performance overhead. By doing so it enables access to those functions on simple HTTP calls converting R from a standalone application to a web service.

User-friendly clients will be developed in Python or as Jupyter notebooks.

**Demonstration of PBPK models that have been exposed as web services:**

The three models (httk, PKSim and user-specified) will be exercised with Imazalil, Theophylline, and Diazepam, respectively.

For each chemical, we will start by identifying relevant human exposures (e.g. from ExpoCast, or published literature) to be used in forward dosimetry. For reverse dosimetry,
we will identify (e.g. from the US NHANES database, or the scientific literature) typical blood or urine concentrations found in humans to be used as input to the exposure dose reconstruction.

The model will be parameterized using user-specified or pre-programmed tabulated physiological data.

For forward dosimetry predictions, each model will be run with the given exposure scenario to predict internal concentrations after 24 hours. Plots of the concentration time-courses will be generated and compared to reference simulations (generated with the stand-alone versions of the same packages).

For reverse dosimetry the model will be run forward iteratively with user set exposures so as to match the input biomarkers (that is: manually invert the model). The external exposure level leading to data-matching biomarker level will be recorded as final estimate.
OUTCOMES

In the Appendix, two tutorials provide descriptions of preliminary implementations of this case study: the first tutorial uses PKSim with theophylline as the chemical of interest. The second tutorial uses httk and Imazalil. Those were demonstrated and received positive feedback during the biokinetics workshop of the OpenTox Euro conference in Basel, 2017, where we had the opportunity to interact with potential users.

The preliminary results of this case-study demonstrates that the OpenRiskNet framework can be used as a central e-platform for the biokinetics community, where the users can publish, share, search and use PBPK models. Additional biokinetic models will be integrated in the future and will be tested in terms of producing successful results (time-concentration profiles). The case study will deliver a User's Manual including a tutorial for this service.
REFERENCES


APPENDIX

Integration of PKsim in OpenRiskNet Infrastructure through the Jaqpot modelling platform

(Presented in OpenRiskNet/OpenTox Euro 2017 Biokinetics Workshop)

The goal of this hands-on workshop is to show how to create a PBPK model using PKSim [1] and expose it as a web service through the Jaqpot modelling platform.

Software requirements: The attendees should download and install in their PCs the Open Systems Pharmacology Suite:  https://github.com/Open-Systems-Pharmacology/Suite/releases/tag/v7.1.0

And the gene expression database

https://github.com/Open-Systems-Pharmacology/Suite/releases/download/v7.1.0/GENEDB_human.mdb

In PKSim select Utilities, then Options, then “Application” and for Human you give the path where the database is stored.

All other necessary files that should be downloaded are available in the google drive:

https://drive.google.com/drive/folders/1wGmqNYI8GnDL_orrE2JqPOAMauHStbP

Assuming that the PBPK model has been saved in xml format (as explained in Powerpoint presentation), the following steps should be followed in Swagger JaqPot API documentation http://test.jaqpot.org:8080/jaqpot/swagger/: 

1) Create a JaqPot dataset containing the physiological parameters of the individual on which the PBPK model was developed. We assume that these parameters (age, height, weight) are included in a csv file testPK.csv
   a. Use the POST /dataset/createDummyDataset method: Choose the testPK.csv file and give any title and description to the produced dataset. In the end of the response body a dataset id is generated.
   b. Use the GET /dataset/{id} method: Copy the dataset id in the relevant field and press Try it out!. In the Request URL the full dataset URI of the produced dataset is shown. Store the URI of the dataset.
      Example: http://test.jaqpot.org:8080/jaqpot/services/dataset/SDea5BEhgvPRTx

2) Produce a JaqPot resource that implements the PBPK model as a web service
   a. Use the POST /biokinetics/train method. Insert the following information in the parameter fields:
      File: Choose the PBPK xml file sim_individual_1_XML.xml
Dataset-uri: Copy the full dataset URI
Title and description: Any title and description is fine
Parameters: {"ageUnit" : ["years"], "individual" : [1], "heightUnit" : ["m"], "weightUnit" : ["kg"], "drug" : ["Theophylline"]}

After pressing the Try it out! button, a task ID is generated in the end of the response body

b. Use the GET /task/{id} method. Copy the task ID in the relevant field and press the Try it out! Button. If the status in the end of the response body is not COMPLETED wait for a few seconds and try again. The model has been created and its complete URI is shown in the response body. You can now share the model with the rest of the world. For the next step store the model id. This is the id after model/ in “result” in the response body

3) Use the model for obtaining drug concentration-time profiles
a. Use the POST /model/{id} method. In the dataset_uri you can use the same dataset URI that was used in the first step. Alternatively, you can create an alternative dataset for a different individual. In the id field copy the id of the produced model. After pressing the Try it out! button a task ID is created.

b. Use the GET /task/{id} method. Copy the task ID in the relevant field and press the Try it out! Button. The dataset with drug concentration –time data has been created. For the next step store the dataset id. This is the id after dataset/ in “result” in the response body

c. Use the GET /dataset/{id} method: Copy the dataset id in the relevant field and press Try it out! In the Response Body, drug concentration values are shown along with corresponding time points.
Integration of httk in OpenRiskNet Infrastructure through the Jaqpot modelling platform

High-Throughput toxicokinetics (httk) is an R package for simulation and analysis of chemical toxicokinetics [2]. httk has been integrated in the Jaqpot platform and will be accessible from the OpenRiskNet infrastructure. The goal of this tutorial is to demonstrate how to obtain toxicokinetics predictions from the httk package using Jaqpot services and functionalities.

Currently the following httk parameters are supported:

1. “chem.name”, which is the chemical name of the compound under study
2. “species”, which is the species of interest (either "Rat", "Rabbit", "Dog", "Mouse", or default "Human")
3. “days”, the length of the simulation in days,
4. “dose”, the amount of the single dose specified (mg/kg/day) with the default value being NULL.

We will use the example of administering Imazalil administered to a human (BW=70kgr) also initializing other basic parameters based on the multiple compartment model introduced in [3]. The compartments used in this model are the gutlumen, gut, liver, kidneys, veins, arteries, lungs, and the rest of the body. The extra compartments include the amounts or concentrations metabolized by the liver and excreted by the kidneys through the tubules.

The following steps should be followed using the Swagger JaqPot API UI at http://jaqpot.org:8080/jaqpot/swagger/:

1) Produce a JaqPot resource that implements the httk model as a web service
   a. Use the POST/biokinetics/httk/createmodel method.

   Insert the following information in the parameter fields:

   - Title and description: Any title and description is fine
   - Parameter string:
     
     
     

     

   Press ‘Try it out!’ In the field response body a json string will appear. There the user can find the task ID.

   b. Use the GET /task/[id] method. Copy the task ID in the relevant field and press the ‘Try it out!’ button. If the status at the end of the response body is not COMPLETED wait for a few seconds and try again. The model has been created and its complete URI is shown in the
The raw model together with the initial parameters of the multiple compartment model, are returned.

2) Use the model for obtaining drug concentration-time profiles

   a. Use the POST biokinetics/httk/model/{id} method in the biokinetics section. Submit model id obtained before and then press the button 'Try it out!'. A task ID is created.

   b. Use the GET /task/{id} method. Copy the task ID in the relevant field and press the Try it out! Button. The dataset with drug concentration –time data has been created. For the next step store the dataset id. This is the id found after dataset/ in the "result" in the response body.

   c. Use the GET /dataset/{id} method: Copy the dataset id in the relevant field and press 'Try it out!'. In the Response Body, drug concentration values are shown along with corresponding time points (in days), and AUC (area under the curve) of the plasma concentration values.

These are the first lines of the result giving concentrations in uM:

<table>
<thead>
<tr>
<th>Substance</th>
<th>Agutlumen</th>
<th>Atubules</th>
<th>Cplasma</th>
<th>Cart</th>
<th>Ckidney</th>
<th>Crest</th>
<th>Cliver</th>
<th>Cgut</th>
<th>Ametabolized</th>
<th>Cven</th>
<th>time</th>
<th>Clung</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2355.4749</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1365.8025</td>
<td>0.1105</td>
<td>13.1333</td>
<td>9.2261</td>
<td>2.8637</td>
<td>277.1605</td>
<td>25 8.6781</td>
<td>0</td>
<td>9.5873</td>
<td>0.0104</td>
<td>9.3056</td>
<td>0.0598</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>791.9496</td>
<td>0.5618</td>
<td>20.3668</td>
<td>14.7784</td>
<td>106.6832</td>
<td>11.1376</td>
<td>356.6631</td>
<td>205.9325</td>
<td>0</td>
<td>9.3056</td>
<td>0.0598</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>459.2054</td>
<td>1.1426</td>
<td>21.3648</td>
<td>15.6043</td>
<td>115.3854</td>
<td>19.338</td>
<td>318.4675</td>
<td>156.002</td>
<td>0</td>
<td>14.8678</td>
<td>0.0208</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>266.2664</td>
<td>1.7291</td>
<td>20.5252</td>
<td>15.0134</td>
<td>111.6516</td>
<td>25.5365</td>
<td>259.9901</td>
<td>120.0944</td>
<td>0</td>
<td>14.6313</td>
<td>0.6817</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>154.3925</td>
<td>2.2885</td>
<td>19.4394</td>
<td>14.2189</td>
<td>105.7722</td>
<td>29.7367</td>
<td>210.7735</td>
<td>96.3627</td>
<td>0</td>
<td>14.1908</td>
<td>0.0521</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>89.5232</td>
<td>2.8195</td>
<td>18.5569</td>
<td>13.5674</td>
<td>100.7951</td>
<td>32.4253</td>
<td>175.7869</td>
<td>81.3744</td>
<td>0</td>
<td>13.5466</td>
<td>0.0625</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>51.9093</td>
<td>3.3285</td>
<td>17.938</td>
<td>13.1087</td>
<td>97.2449</td>
<td>34.0891</td>
<td>152.7805</td>
<td>72.164</td>
<td>0</td>
<td>13.0947</td>
<td>0.0729</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>30.0992</td>
<td>3.8225</td>
<td>17.5331</td>
<td>12.8081</td>
<td>94.9023</td>
<td>35.0962</td>
<td>138.2902</td>
<td>66.602</td>
<td>0</td>
<td>12.7992</td>
<td>0.0833</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>17.4528</td>
<td>4.3068</td>
<td>17.2782</td>
<td>12.6186</td>
<td>93.42</td>
<td>35.6962</td>
<td>129.3984</td>
<td>63.2808</td>
<td>0</td>
<td>12.6131</td>
<td>0.0938</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Other examples:

{"chem.name":["bisphenol A"],"species":["Rat"],"days":[10],"dose":[10]}

{"chem.name":["imazalil"],"species":["Rat"],"days":[10],"dose":[10]}

{"chem.name":["Acetochlor"],"species":["Rat"],"days":[10],"dose":[10]}
Integration of custom made PBPK models in the Jaqpot modelling platform

Besides the integration of httk and PKSim, we offer the option to the users to create, upload and share a custom-made PBPK model through the Jaqpot infrastructure.

An example is the PBPK model described in [4], which describes the biokinetics of diazepam in humans.

The model is currently accessible through the local deployment of Jaqpot in the following URL: [https://app.jaqpot.org/model/ecTo8uVZb7AF9KBXp0sp](https://app.jaqpot.org/model/ecTo8uVZb7AF9KBXp0sp).

Under the Overview tab, detailed information about the model are provided:

Under the Data tab, all independent (weight, gender, infusion time, dose and initial concentration in all compartments) and dependent variables (concentrations in all compartments: adipose, muscle, brain, skin, gonads, kidneys, heart, the rest of the body, stomach, liver, arterial blood, lungs, intestines, venous blood) are defined.

In the Predict/Validate tab the user can simulate a scenario. As an example, the model is asked to simulate the concentration-time profiles produced after administering 5mg of
diazepam to a male weighted 70Kg. The infusion time is 0.1 hours and all initial concentrations are assumed equal to 0.