Jaqpot - An open-source web platform for creating, using, testing and sharing predictive models in nano-informatics

Haralambos (Harry) Sarimveis
Professor
School of Chemical Engineering
National Technical University of Athens, Greece
E-mail: hsarimv@central.ntua.gr
https://www.chemeng.ntua.gr/labs/control_lab/sarimveis.html

Unit of Process Control and Informatics

- 1 Professor, 2 senior researchers, 3 post-docs, 6 Ph.D. students, 3 software engineers, many graduate and undergraduate students.
- Research Interests: mathematical modelling, data-driven methods, machine learning, computational intelligence, mathematical programming
- Applications: process control, production optimization, modelling in the chemical industry, **cheminformatics, nanoinformatics**

Outline

1. Development of Jaqpot
2. Technical Specifications
3. Predictive nanoQSAR modelling
4. Data management
5. Biokinetics – PBPK modelling
6. Image Processing
7. Other Jaqpot functionalities
8. Conclusions

Jaqpot at a glance

Jaqpot is a user-friendly web-based e-infrastructure containing many data analysis and modelling microservices integrated under a common API.

The Jaqpot infrastructure allows for:

- Preprocessing data
- Computing descriptors from raw data (such as electronic images)
- Creating, validating, storing and sharing datasets and predictive statistical and machine learning models
- Generating reports in standard formats.

Developent of Jaqpot

- Original development during the OpenTox project (2008—2011)

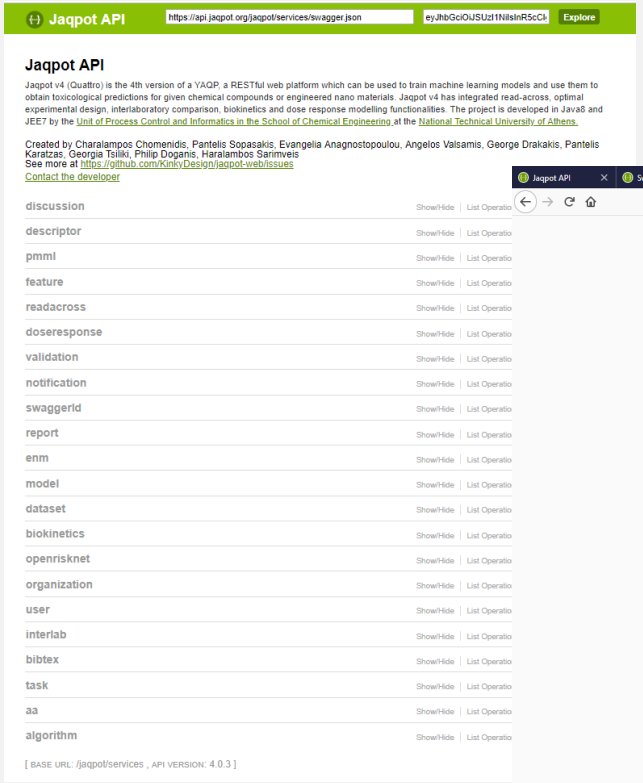
- Main modelling tool in the eNanoMapper project (2014-2017)

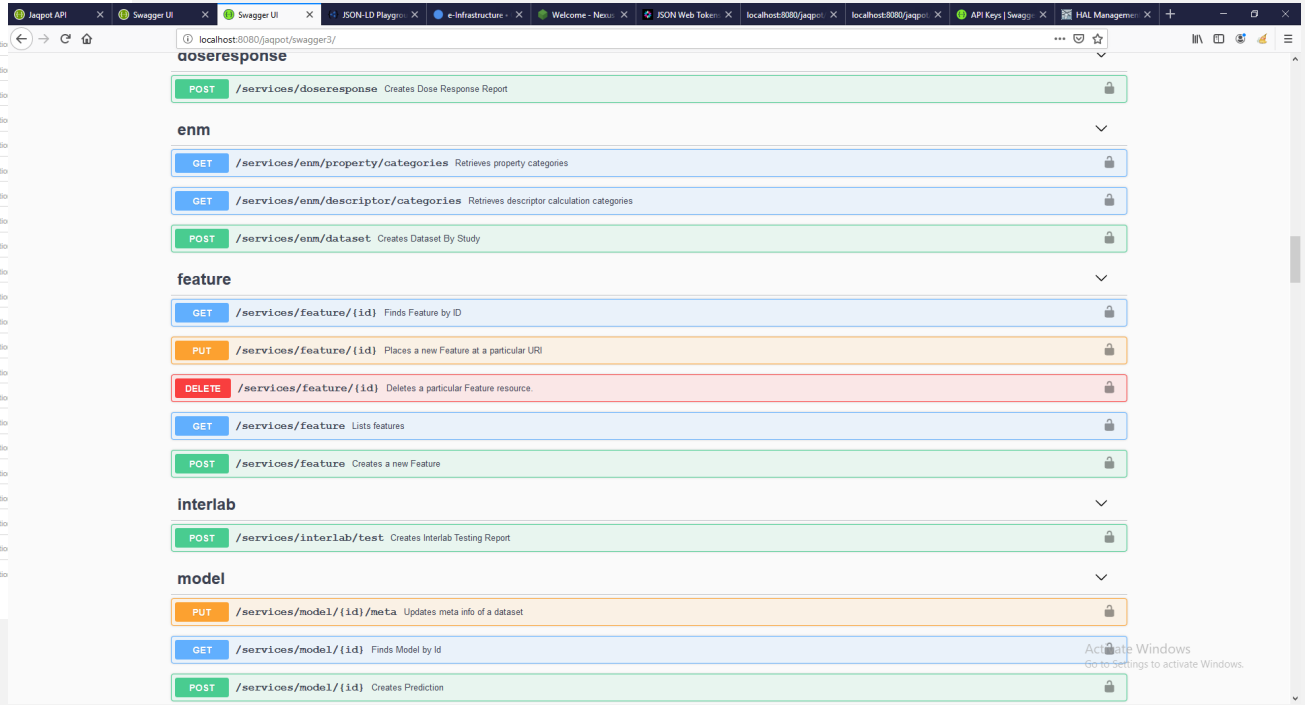
- Part of the OpenRiskNet open e-infrastructure (2017-2019)

- Major Component of the NanoCommons community framework and infrastructure (2018-2021)

- Provides infrastructure and modelling tools to the NanoSolveIT e-platform (2019-2023)

Application Programming Interfaces (Swagger Documentation)



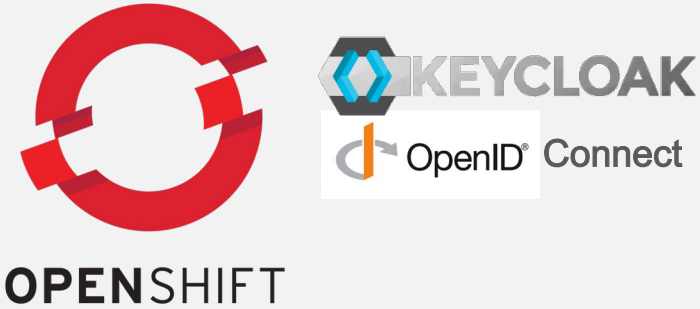


Infrastructure

Docker is used to run software packages called "containers". Containers are isolated from each other and bundle their own application, tools, libraries and configuration files;

An open-source system for automating deployment, scaling, and management of containerized applications

OpenShift is a multifaceted, open source container application platform from Red Hat Inc. for the development, deployment and management of applications.



An open source software product to allow OpenID Connect single sign-on with Identity Management and Access Management aimed at modern applications and services.

nanoQSA(P)R

The development of a **mathematical relationship** y=f(X), that predicts an end-point (property, toxicity index) as a function of one or more descriptors (phys-chem properties, computational descriptors, image descriptors, omics data)

OECD principles:

1. A defined **end-point**
2. The use of an **unambiguous algorithm**
3. A defined **domain of applicability**
4. Appropriate measures of **goodness-of-fit**, **robustness** and **predictive ability**
5. A **mechanistic interpretation**, if possible

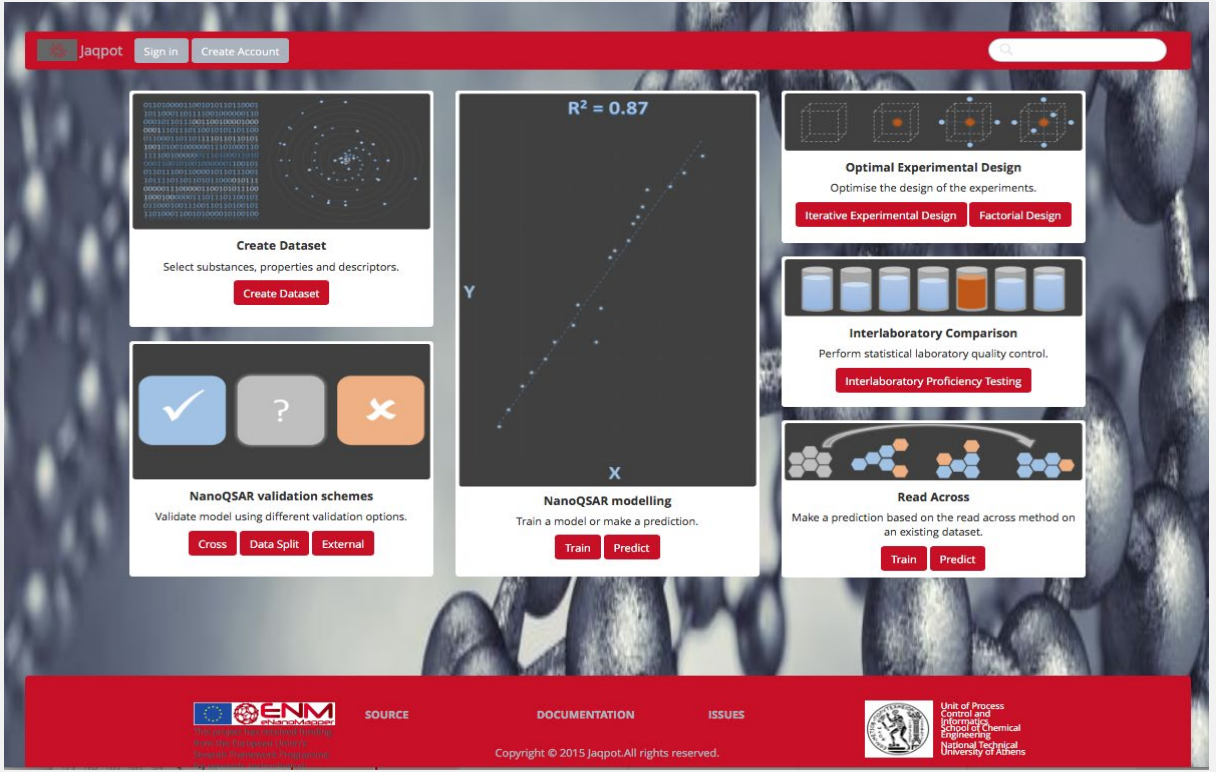
Reporting formats developed by Joint Research Center (JRC):

QSAR Model Reporting Format **(QMRF)**
QSAR Prediction Reporting Format (**QPRF**)

Guidance Document on the Validation of (Quantitative) Structure-Activity Relationship [(Q)SAR] Models; OECD SERIES ON TESTING AND ASSESSMENT. (OECD, 2007), ENV/JM/MONO(2007)2

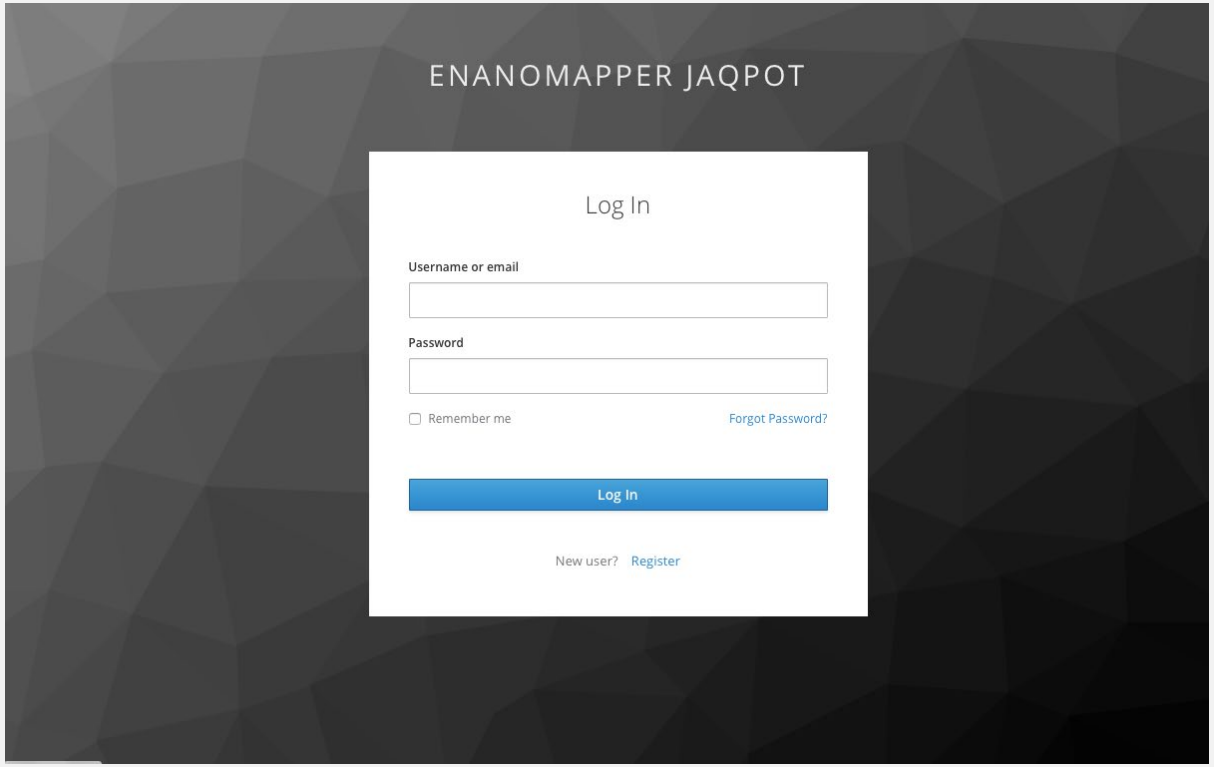
Jaqpot 4

http://www.jaqpot.org/

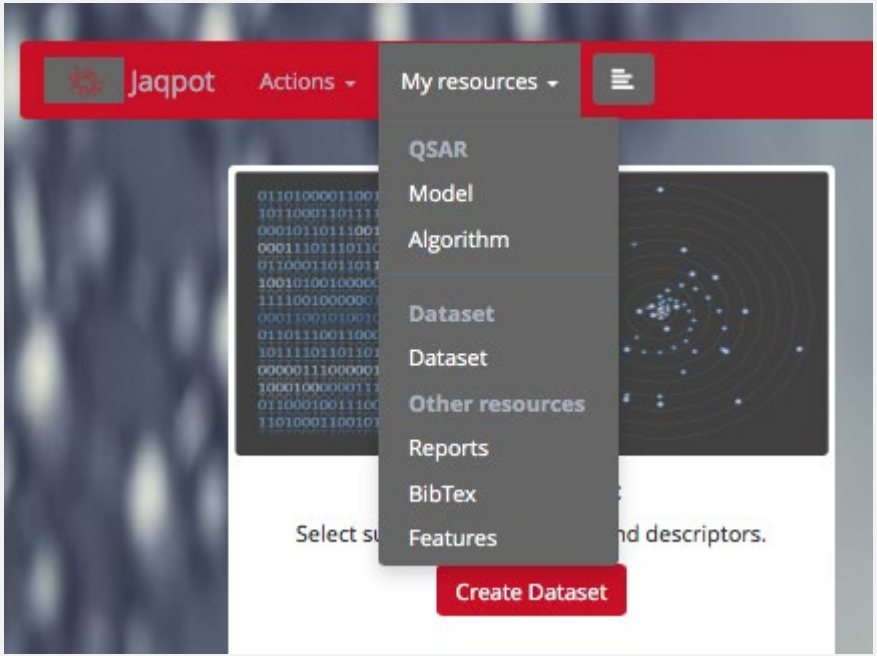


Chomenidis C, Drakakis, G, Tsiliki G, Anagnostopoulou E, Valsamis A, Doganis P, Sopasakis P, Sarimveis, H, (2017) Jaqpot Quattro: A Novel Computational Web Platform for Modeling and Analysis in Nanoinformatics, Journal of Chemical Information and Modeling, 57(9), 2161-2172.

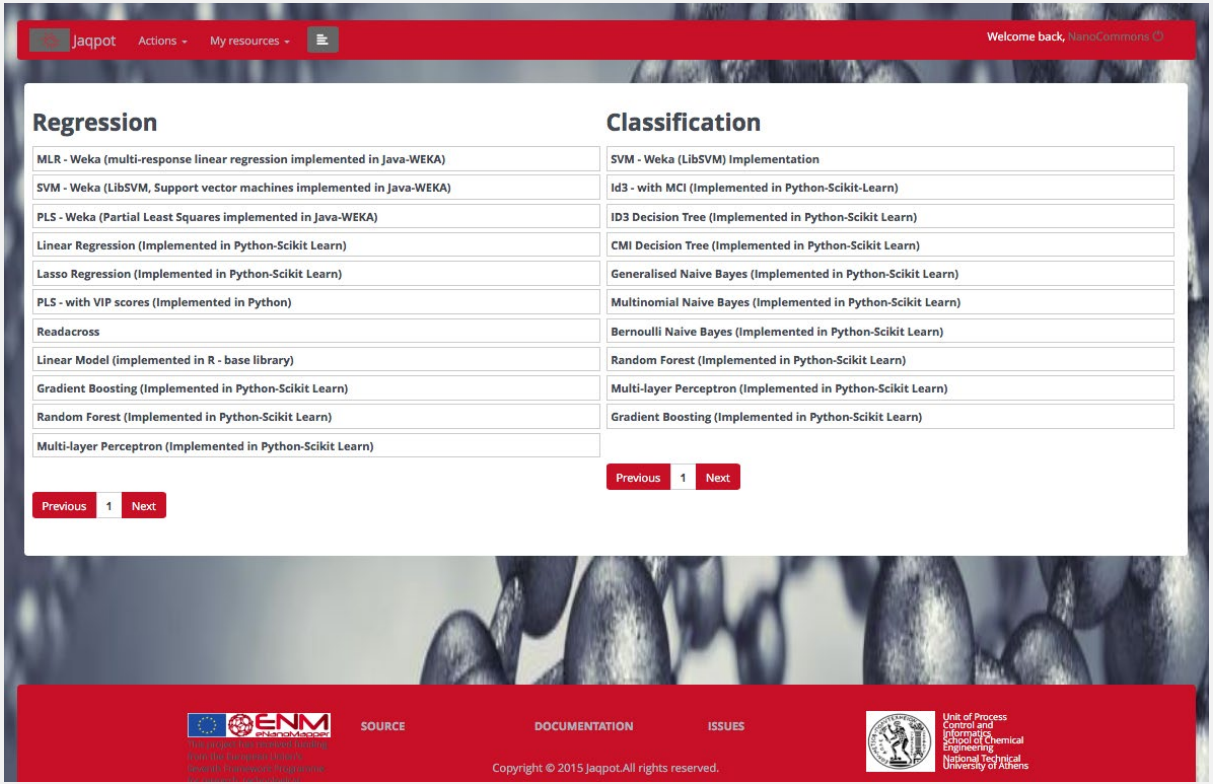
Create new account



My resources



Algorithms


Jaqpot
Actions
My resources
Welcome back, NanoCommons
Regression
MLR - Weka (multi-response linear regression implemented in Java-WEKA)
SVM - Weka (LibSVM, Support vector machines implemented in Java-WEKA)
PLS - Weka (Partial Least Squares implemented in Java-WEKA)
Linear Regression (Implemented in Python-Scikit Learn)
Lasso Regression (Implemented in Python-Scikit Learn)
PLS - with VIP scores (Implemented in Python)
Readacross
Linear Model (implemented in R - base library)
Gradient Boosting (Implemented in Python-Scikit Learn)
Random Forest (Implemented in Python-Scikit Learn)
Multi-layer Perceptron (Implemented in Python-Scikit Learn)
Previous 1 Next
Classification
SVM - Weka (LibSVM) Implementation
Id3 - with MCI (Implemented in Python-Scikit-Learn)
ID3 Decision Tree (Implemented in Python-Scikit Learn)
CMI Decision Tree (Implemented in Python-Scikit Learn)
Generalised Naive Bayes (Implemented in Python-Scikit Learn)
Multinomial Naive Bayes (Implemented in Python-Scikit Learn)
Bernoulli Naive Bayes (Implemented in Python-Scikit Learn)
Random Forest (Implemented in Python-Scikit Learn)
Multi-layer Perceptron (Implemented in Python-Scikit Learn)
Gradient Boosting (Implemented in Python-Scikit Learn)
Previous 1 Next
SOURCE
DOCUMENTATION
ISSUES
Copyright © 2015 Jaqpot.All rights reserved.
Unit of Process Control and Informatics
School of Chemical Engineering
National Technical University of Athens

Ontological annotation of algorithms

	Category / URI	Description	OpenTox Ontological Classes
	Preparation-additional		
1	http://jaqpot.org:8080/jaqpot/services/algorithm/scaling	Scaling	"ot:Algorithm", "ot:Scaling", "ot:Transformation"
	WEKA (Java)		
1	http://jaqpot.org:8080/jaqpot/services/algorithm/weka-mlr	MLR - Weka (multi-response linear regression implemented in Java-WEKA)	"ot:Algorithm", "ot:Regression", "ot:SupervisedLearning"
	Python		
1	http://jaqpot.org:8080/jaqpot/services/algorithm/python-id3-mci	Id3 - with MCI (Implemented in Python-Scikit-Learn)	"ot:Algorithm", "ot:Classification", "ot:SupervisedLearning"

OpenTox Algorithm Ontology http://old.opentox.org/data/documents/development/RDF%20files/AlgorithmTypes

Model training

Example

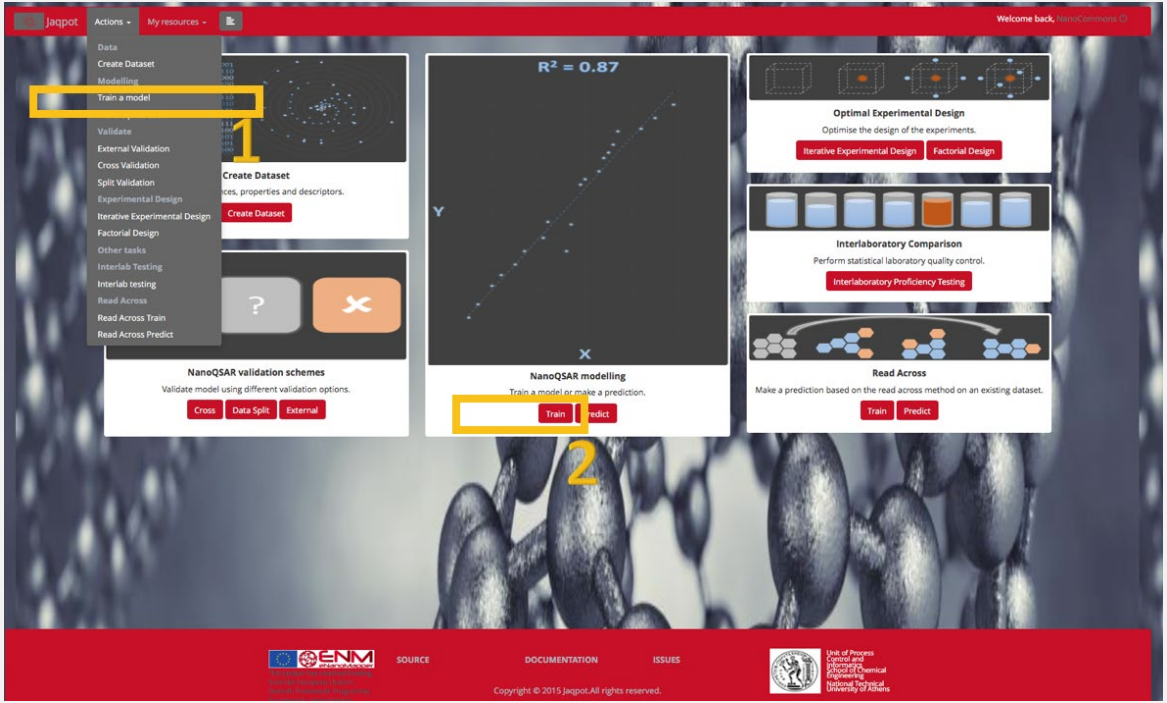
Modelling the Solubility of C60 fullerene in various solvents (Gharageizi et.al., 2008).

The full dataset of 124 solvents has been uploaded to Jaqpot and can be viewed in the URI http://www.jaqpot.org/data_detail?name=nGF3G5SBo4wk5h

Training dataset of 93 solvents

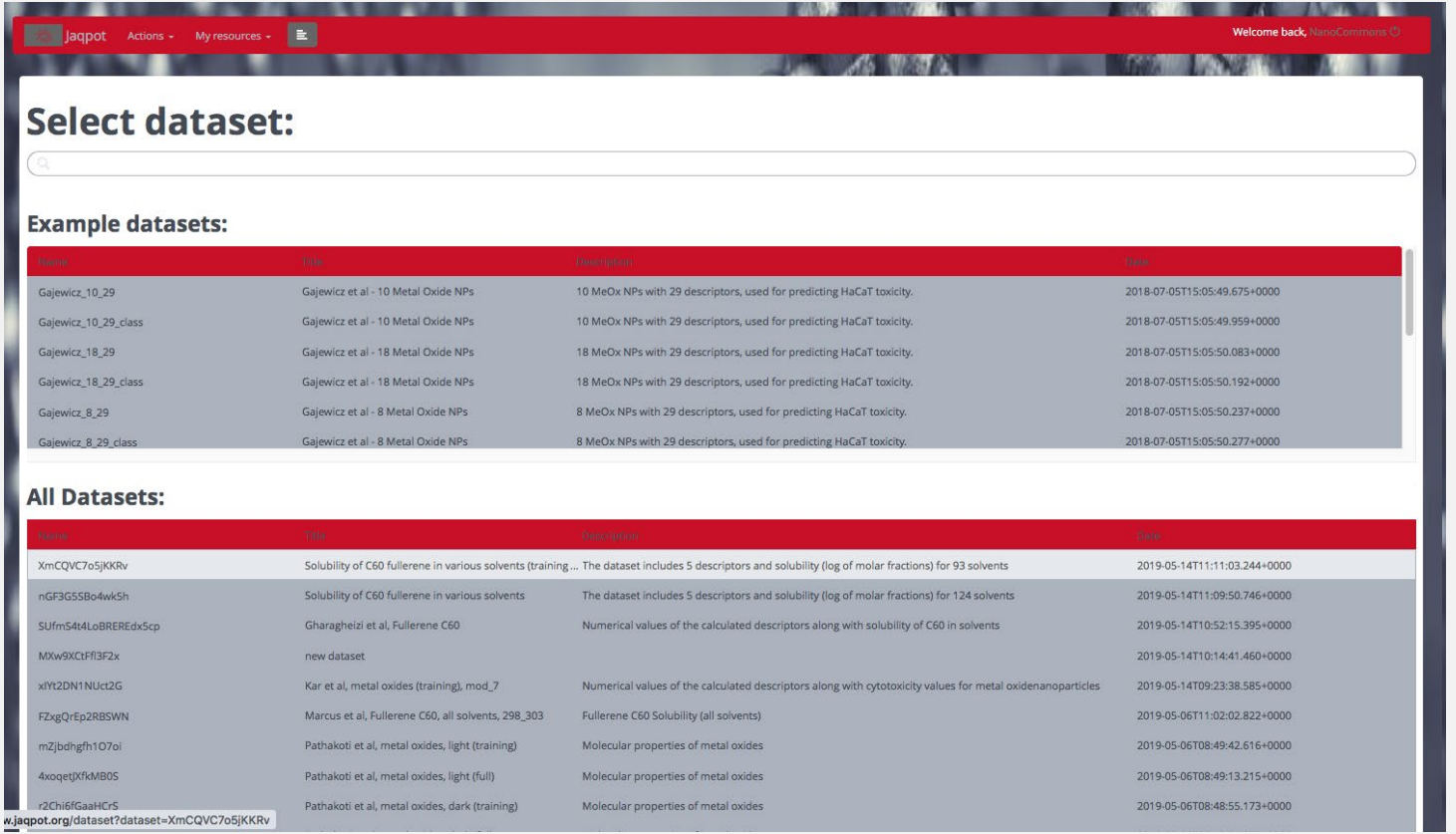
http://www.jaqpot.org/data_detail?name=XmCQVC7o5jKKRv)

Test dataset of 31 solvents (http://www.jaqpot.org/data_detail?name=3UbgEJPIdT2Ovs)

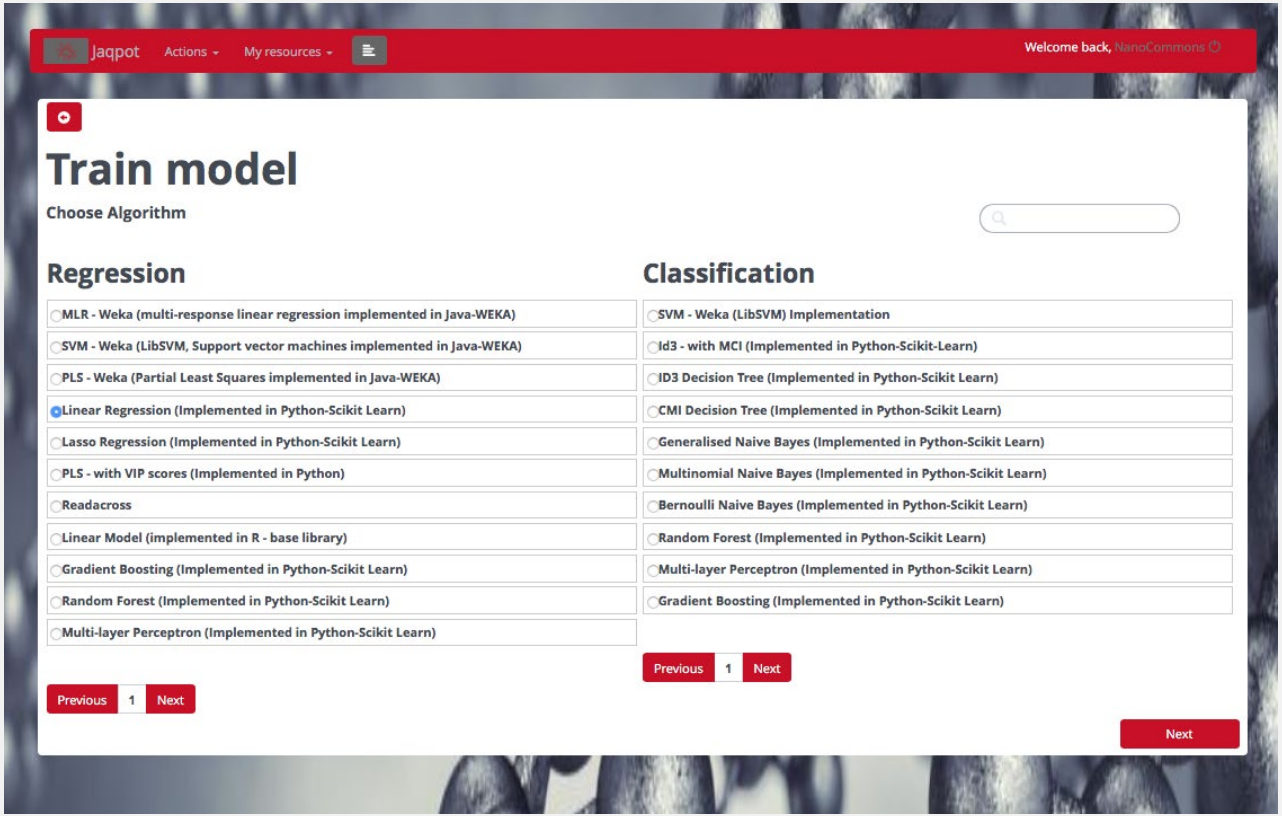


Farhad Gharagheizi & Reza Fareghi Alamdari (2008) A Molecular-Based Model for Prediction of Solubility of C60 Fullerene in Various Solvents, Fullerenes, Nanotubes, and Carbon Nonstructures, 16:1, 40-57, DOI: 10.1080/15363830701779315

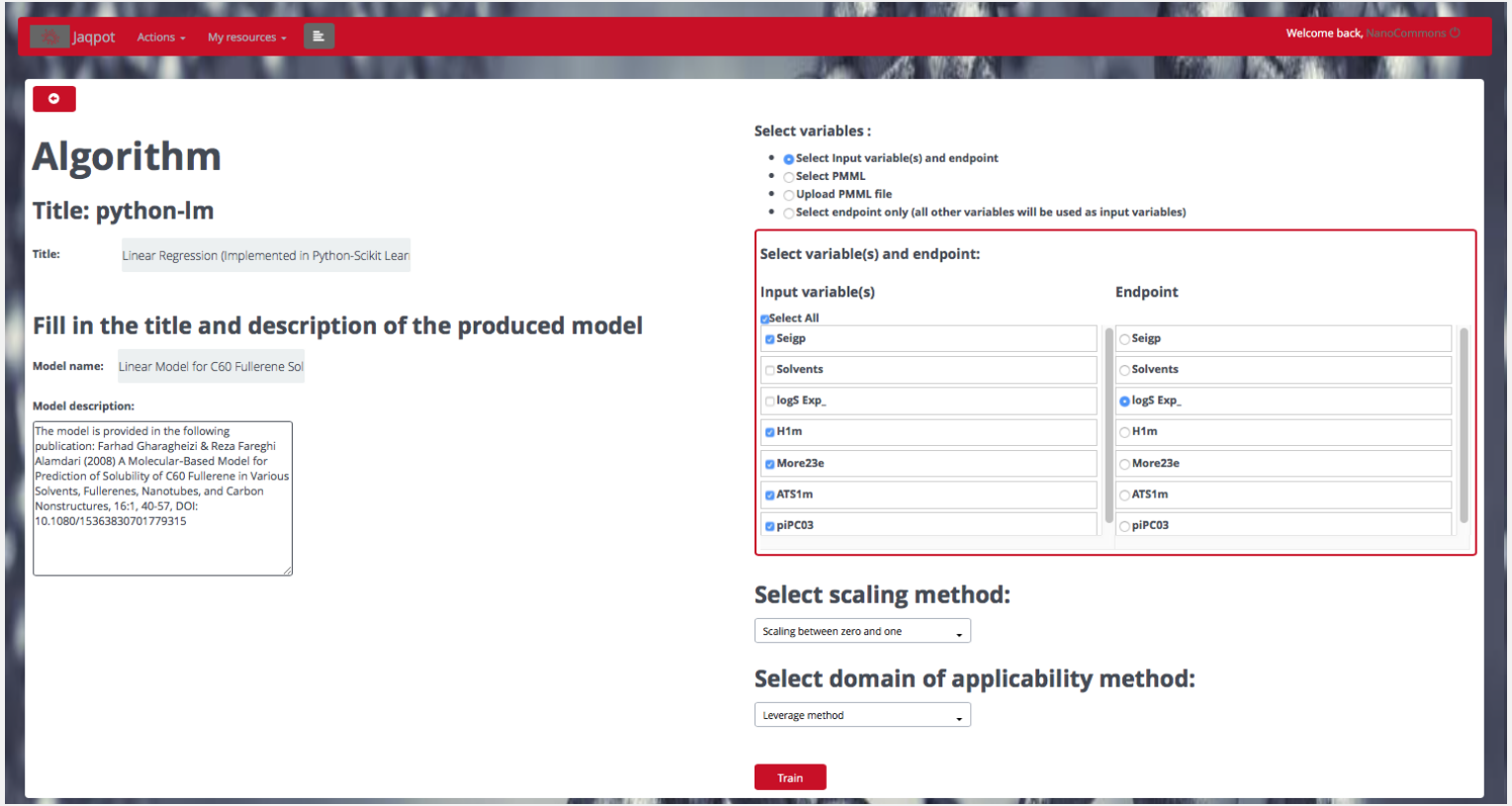
Model training - Dataset selection



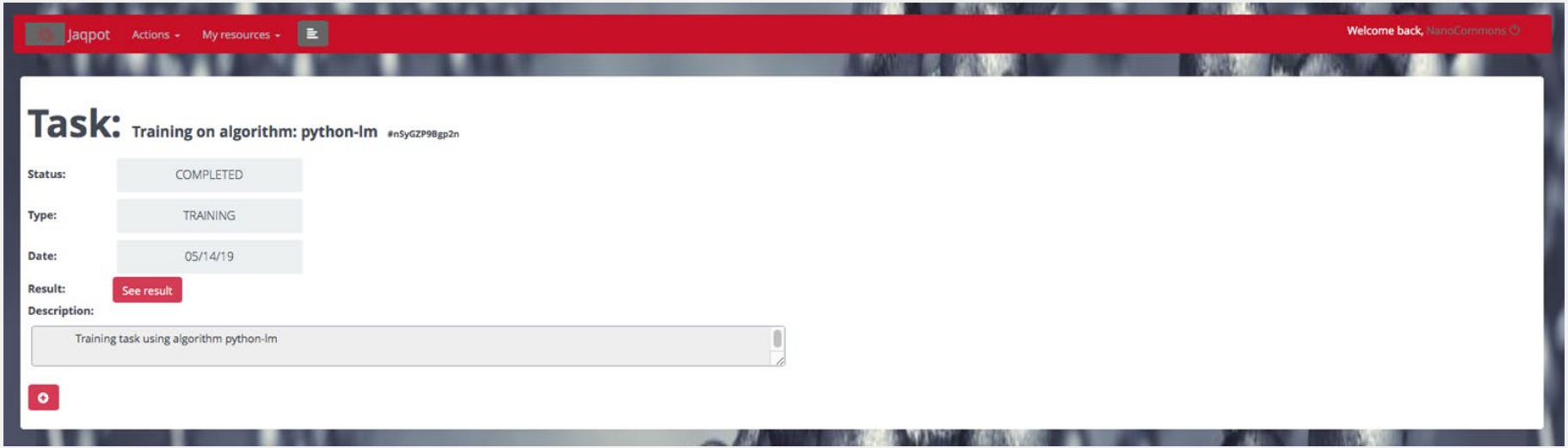
Model training - Choice of algorithm



Model training – Variable and parameters selection

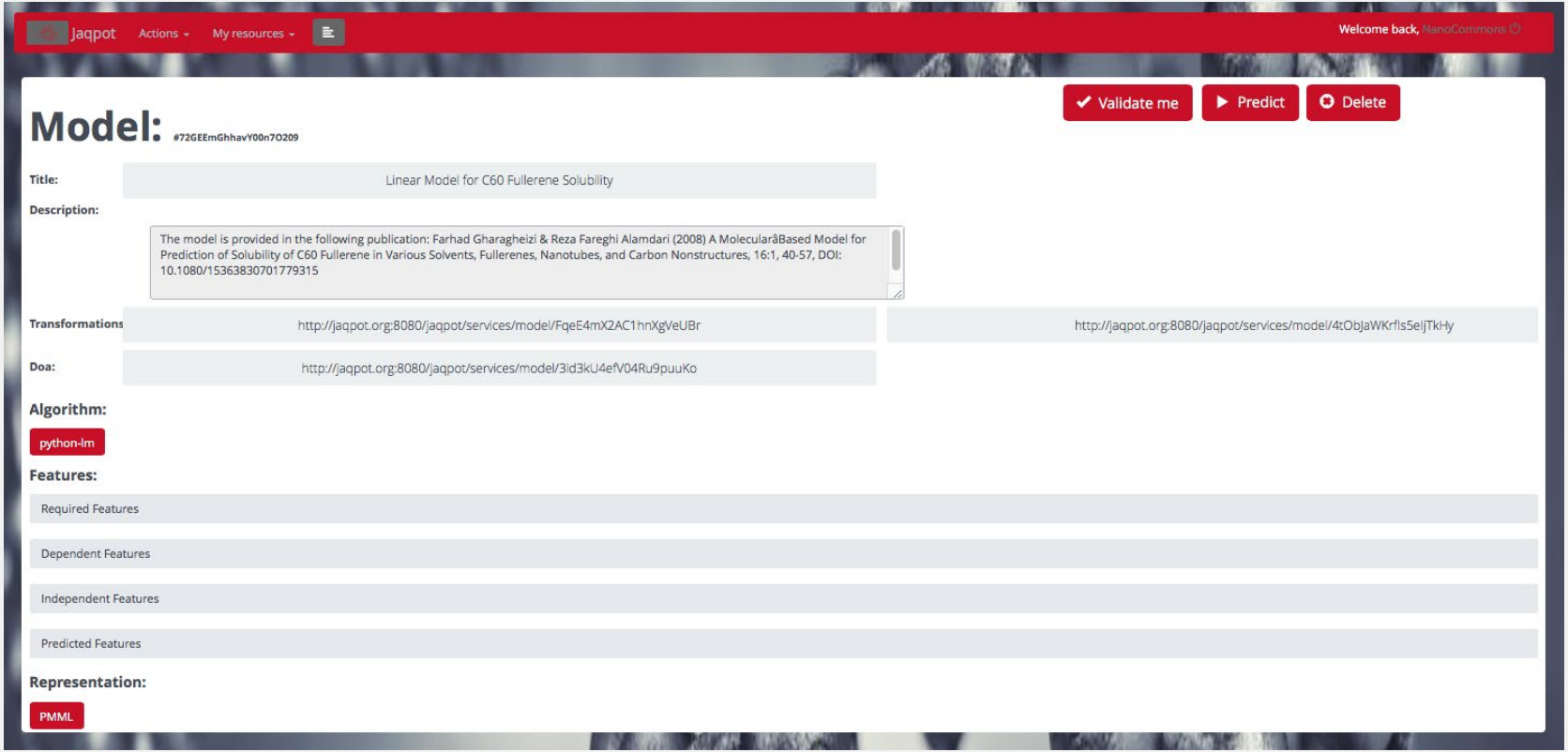


Task management



Model webpage

URI of model: http://www.jaqpot.org/m_detail?name=72GEEmGhhavY00n7O209

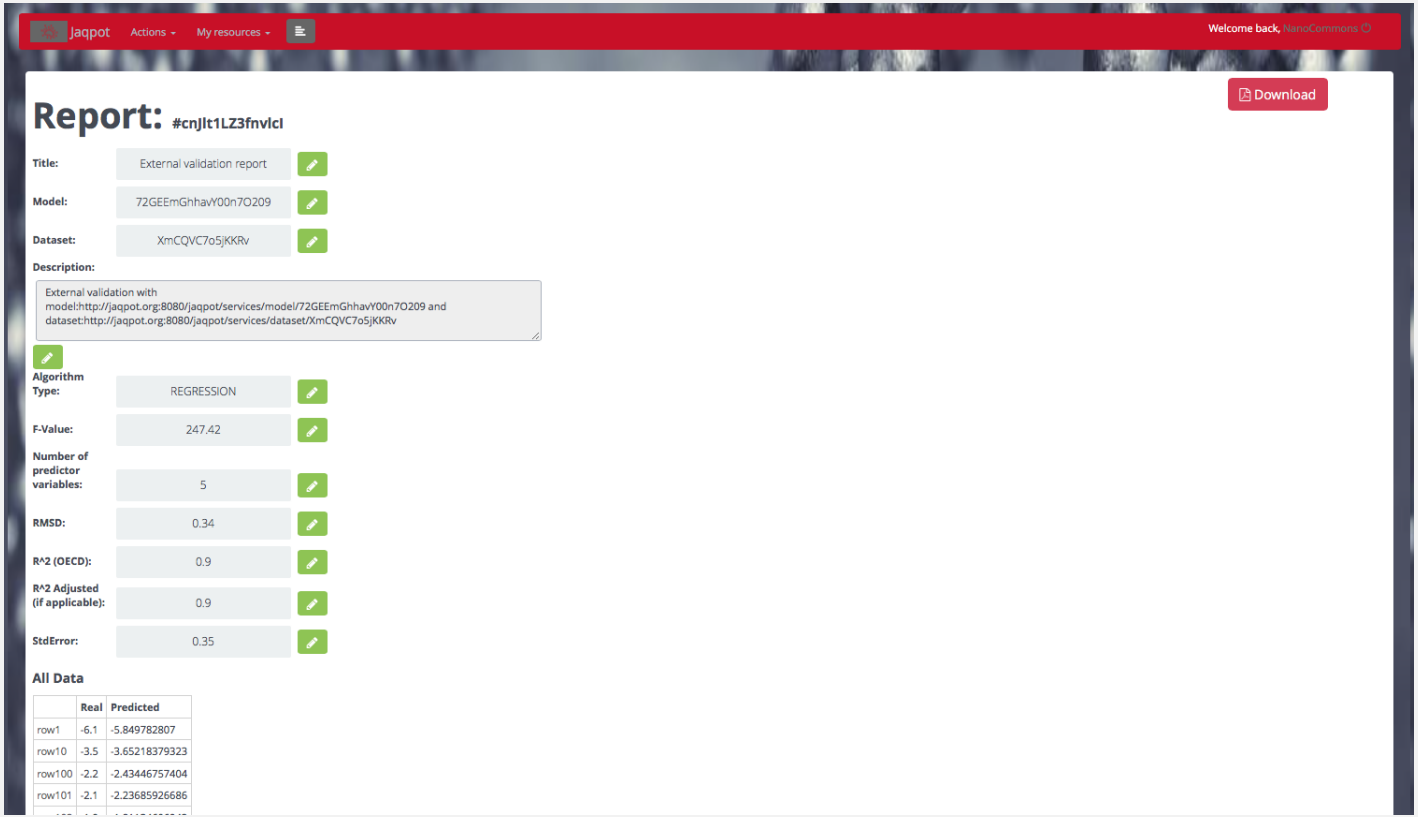


External validation of models

	piPC03	ATS1m	Seigp	More23e	H1m	logS Exp_
1	1,609	2,473	0,578	0,025	0,927	-4
2	1,609	1,763	-1,8	0,022	0,328	-5,3
3	3,426	2,398	0	-1,159	0,449	-3,4
4	1,099	1,674	-1,2	-0,449	0,192	-5,9
5	1,099	2,792	0,846	-0,614	1,495	-4,2

Validation report

http://www.jaqpot.org/report?name=cnJlt1LZ3fnvlcl



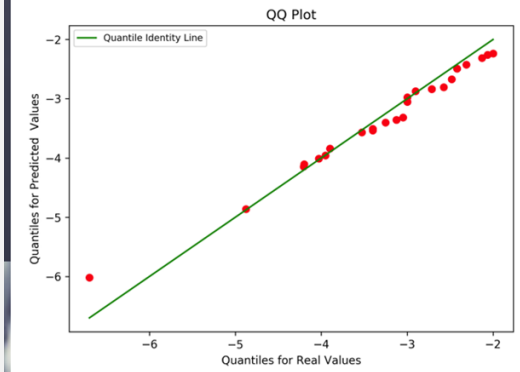
Validation report

All Data

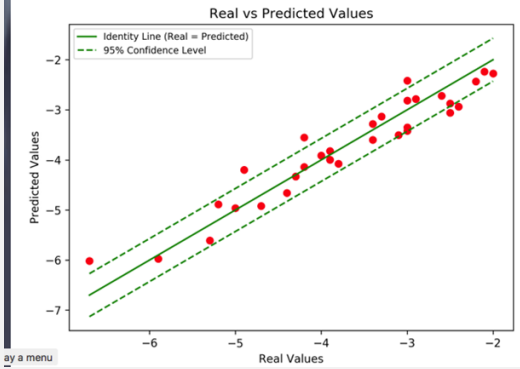
	Real	Predicted
compound1	-4	-3.91355789467
compound10	-2.6	-2.72064673994
compound11	-4.9	-4.19945022239
compound12	-5	-4.96443038285
compound13	-2.1	-2.23685926686
compound14	-3	-3.34976177746
compound15	-2	-2.27234329025
compound16	-3.9	-3.82107851451
compound17	-5.2	-4.88772442914
compound18	-4.2	-3.5504783112
compound19	-6.7	-6.01794584696
compound2	-5.3	-5.61018322885
compound20	-4.3	-4.33212597512
compound21	-2.5	-3.05827270109
compound22	-3.3	-3.13183407097
compound23	-2.4	-2.93903145294
compound24	-3.8	-4.07882245078
compound25	-3	-2.81223578464
compound26	-4.4	-4.65814301486
compound27	-4.7	-4.91962777437
compound28	-2.5	-2.87194521299
compound29	-3.9	-3.99928281415
compound3	-3.4	-3.28038430934
compound30	-2.2	-2.43446757404
compound31	-3	-3.41838538257
compound4	-5.9	-5.97659581681
compound5	-4.2	-4.14080958734
compound6	-3.1	-3.5059080608
compound7	-2.9	-2.78268289953
compound8	-3.4	-3.59838096199
compound9	-3	-2.41555106252

Validation report

QQ Plot



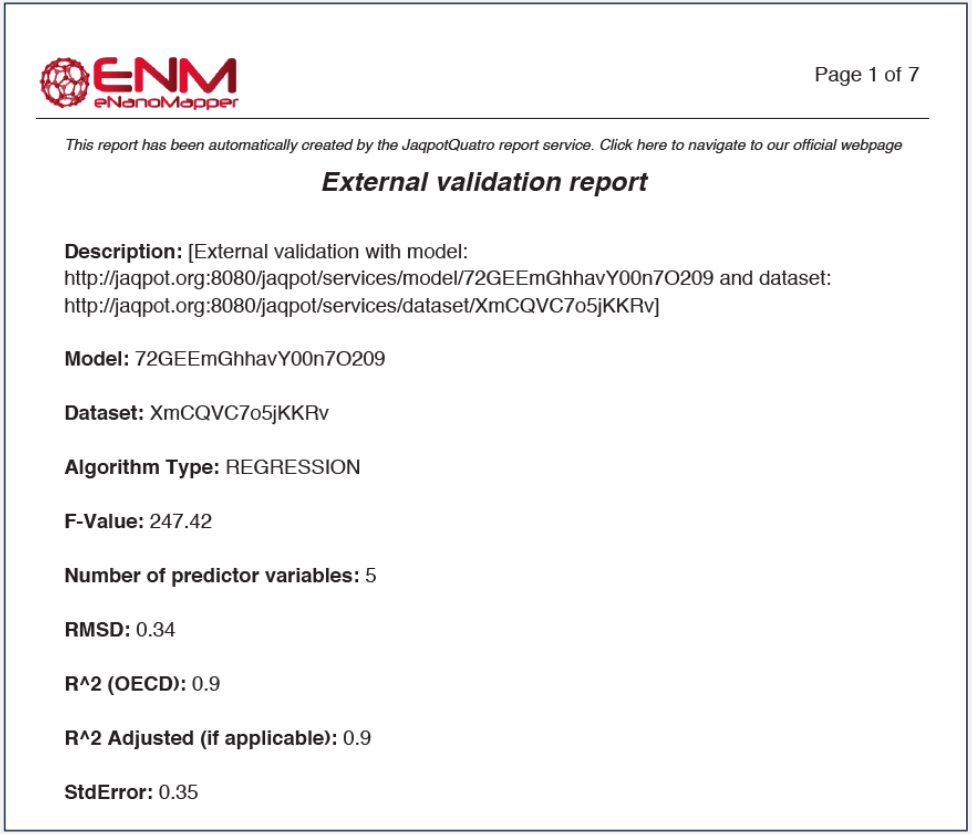
Real Vs Predicted



ay a menu

Validation report

The report is downloadable as PDF



eNM eNanoMapper

This report has been automatically created by the JaqpotQuatro report service. Click here to navigate to our official webpage

External validation report

Description: [External validation with model: http://jaqpot.org:8080/jaqpot/services/model/72GEEmGhhavY00n7O209 and dataset: http://jaqpot.org:8080/jaqpot/services/dataset/XmCQVC7o5jKKRv]

Model: 72GEEmGhhavY00n7O209

Dataset: XmCQVC7o5jKKRv

Algorithm Type: REGRESSION

F-Value: 247.42

Number of predictor variables: 5

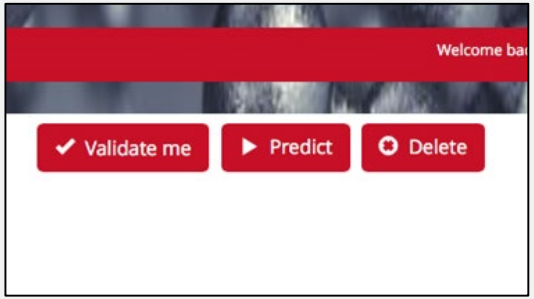
RMSD: 0.34

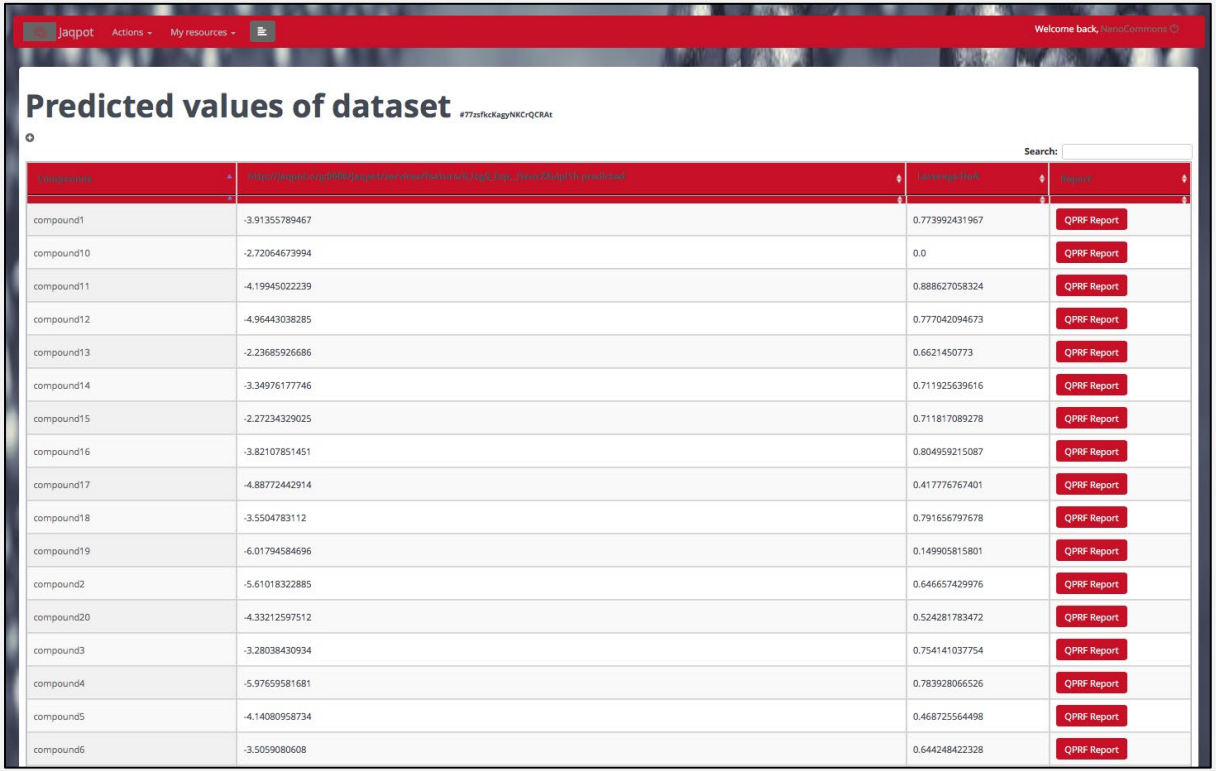
R^2 (OECD): 0.9

R^2 Adjusted (if applicable): 0.9

StdError: 0.35

Model predictions



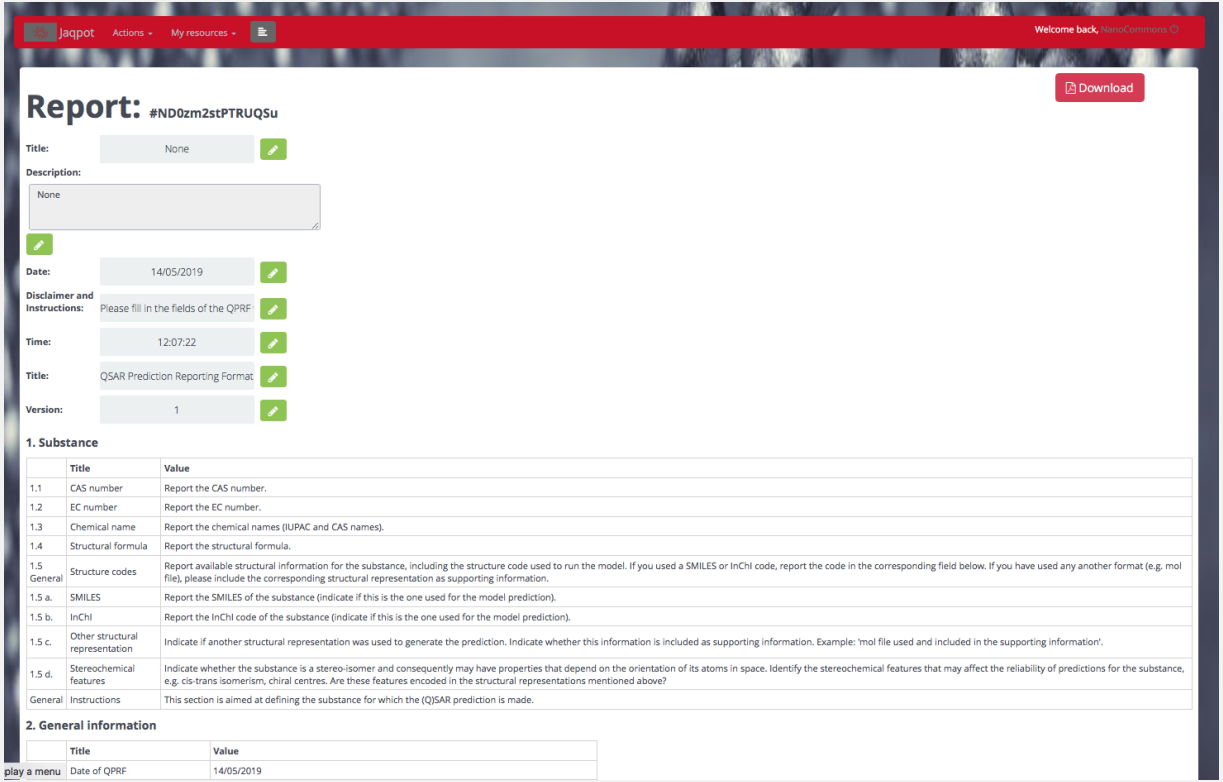


compound1	-3.91355789467	0.773992431967	QPRF Report
compound10	-2.72064673994	0.0	QPRF Report
compound11	-4.19945022239	0.888627058324	QPRF Report
compound12	-4.96443038285	0.77704094673	QPRF Report
compound13	-2.23685926686	0.6621450773	QPRF Report
compound14	-3.34976177746	0.711925639616	QPRF Report
compound15	-2.27234329025	0.711817089278	QPRF Report
compound16	-3.82107851451	0.804959215087	QPRF Report
compound17	-4.88772442914	0.417776767401	QPRF Report
compound18	-3.5504783112	0.791656797678	QPRF Report
compound19	-6.01794584696	0.149905815801	QPRF Report
compound2	-5.61018322885	0.646657429976	QPRF Report
compound20	-4.33212597512	0.524281783472	QPRF Report
compound3	-3.28038430934	0.754141037754	QPRF Report
compound4	-5.97659581681	0.783928066526	QPRF Report
compound5	-4.14080958734	0.468725564498	QPRF Report
compound6	-3.5059080608	0.644248422328	QPRF Report

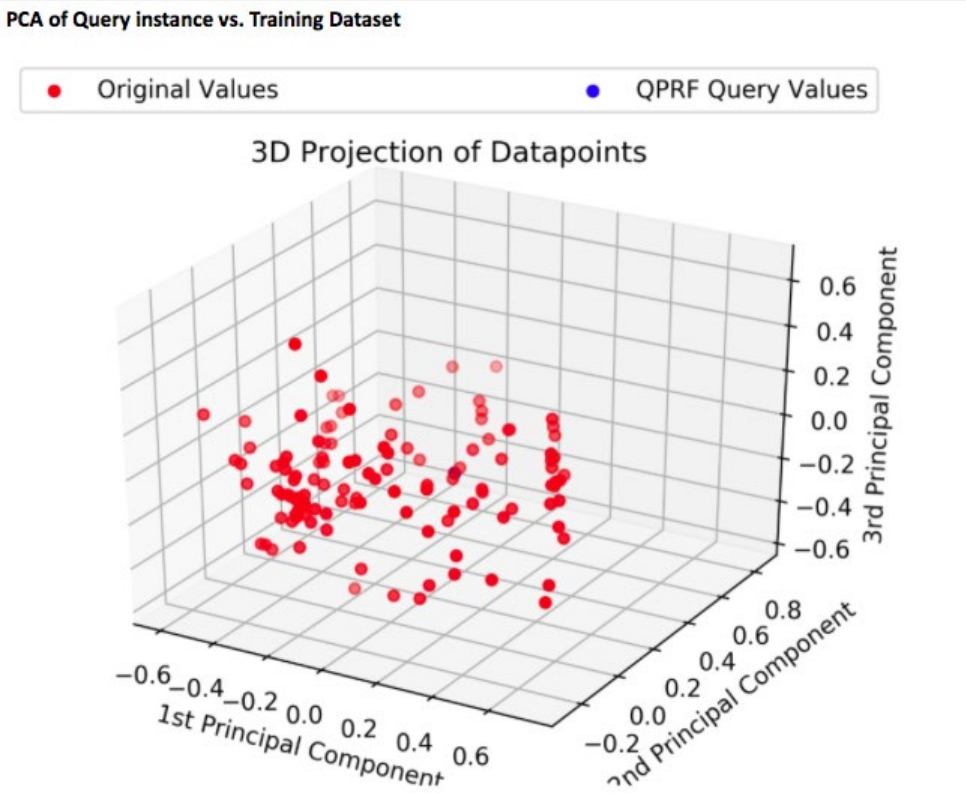
QPRF report

The QPRF (QSAR prediction reporting format) report generated by Jaqpot contains all the fields required by the OECD guidelines, namely:

- Substance
  - Contains information such as CAS and EC numbers, SMILES, InChi, etc.
- General
  - Information such as date and creator name and email.
- Prediction
  - Biological endpoint, variables, model used, DoA, etc
- Adequacy
  - Optional field, containing regulatory purpose, conclusion etc

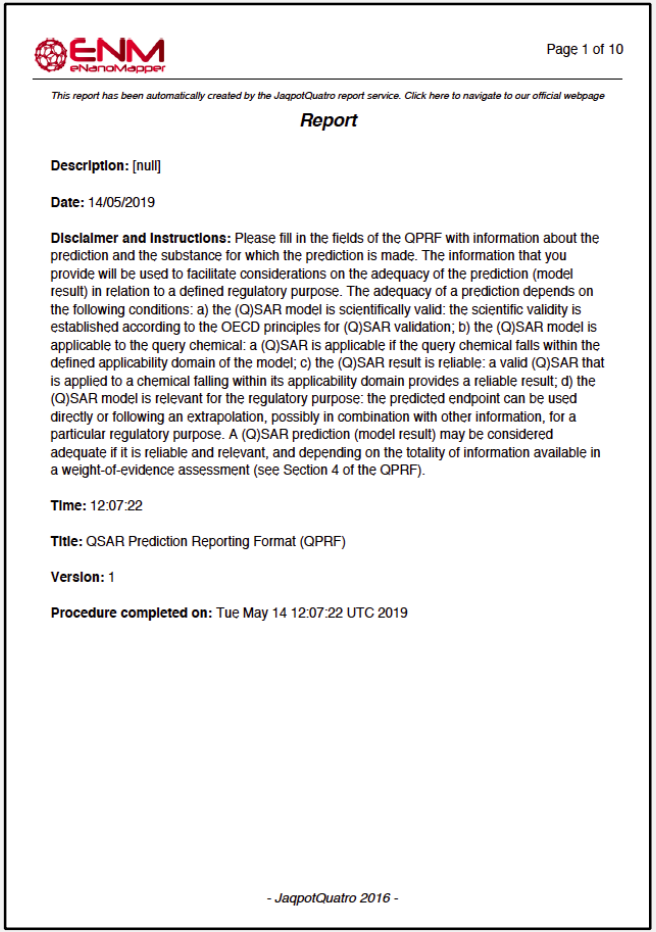


QPRF report


PCA of Query instance vs. Training Dataset
Original Values
QPRF Query Values
3D Projection of Datapoints
3rd Principal Component
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
1st Principal Component
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
2nd Principal Component
0.8
0.6
0.4
0.2
0.0
−0.2

QPRF report

The report is downloadable as PDF



ENM eNanoMapper

This report has been automatically created by the JaqpotQuatro report service. Click here to navigate to our official webpage

Report

Description: [null]

Date: 14/05/2019

Disclaimer and Instructions: Please fill in the fields of the QPRF with information about the prediction and the substance for which the prediction is made. The information that you provide will be used to facilitate considerations on the adequacy of the prediction (model result) in relation to a defined regulatory purpose. The adequacy of a prediction depends on the following conditions: a) the (Q)SAR model is scientifically valid: the scientific validity is established according to the OECD principles for (Q)SAR validation; b) the (Q)SAR model is applicable to the query chemical: a (Q)SAR is applicable if the query chemical falls within the defined applicability domain of the model; c) the (Q)SAR result is reliable: a valid (Q)SAR that is applied to a chemical falling within its applicability domain provides a reliable result; d) the (Q)SAR model is relevant for the regulatory purpose: the predicted endpoint can be used directly or following an extrapolation, possibly in combination with other information, for a particular regulatory purpose. A (Q)SAR prediction (model result) may be considered adequate if it is reliable and relevant, and depending on the totality of information available in a weight-of-evidence assessment (see Section 4 of the QPRF).

Time: 12:07:22

Title: QSAR Prediction Reporting Format (QPRF)

Version: 1

Procedure completed on: Tue May 14 12:07:22 UTC 2019

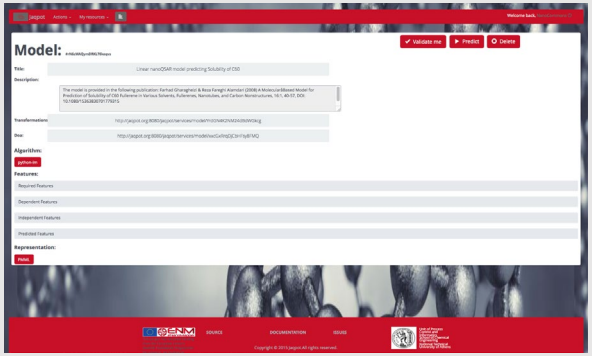
- JaqpotQuatro 2016 -

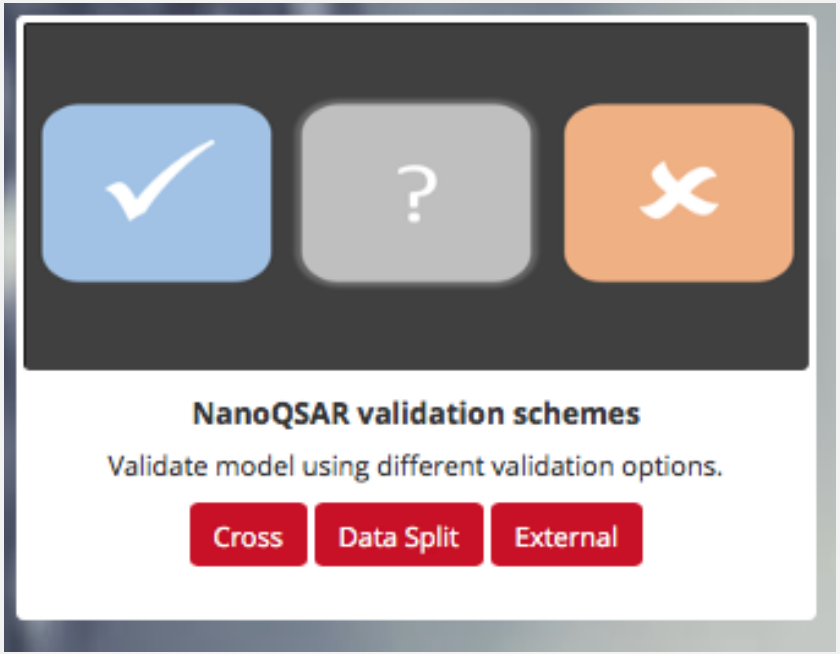
Other model validation options

Options:

- Cross :performs Cross Validation
- Data Split :splits data into training and test datasets according to a split ratio
- External :validation with an external dataset

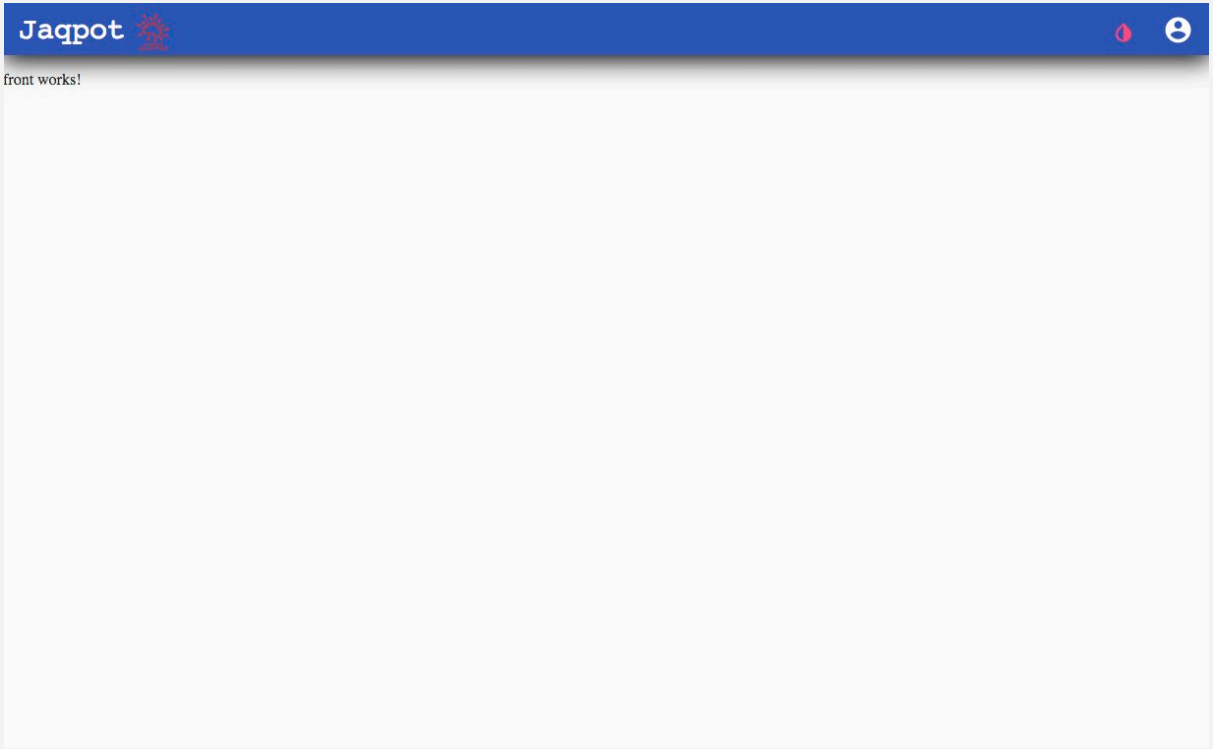
Offered in model webpage



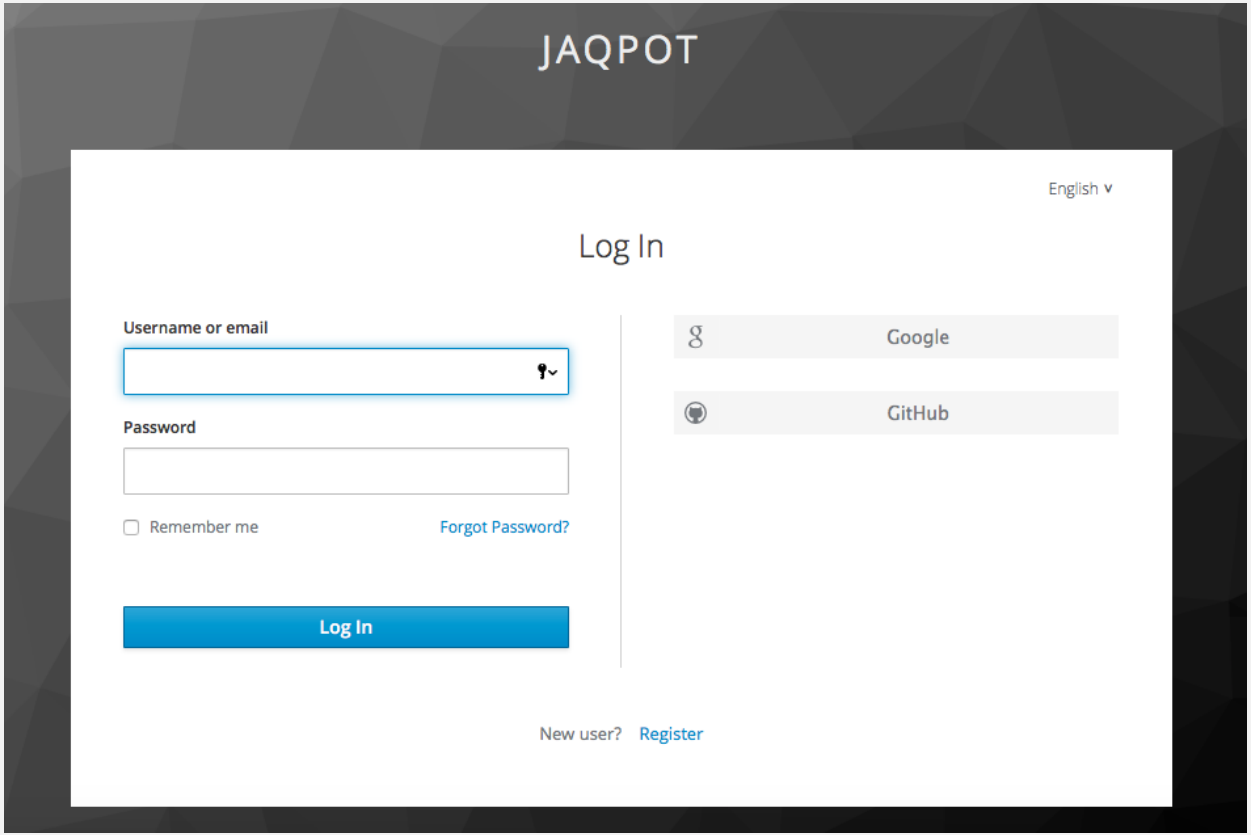


Jaqpot 5

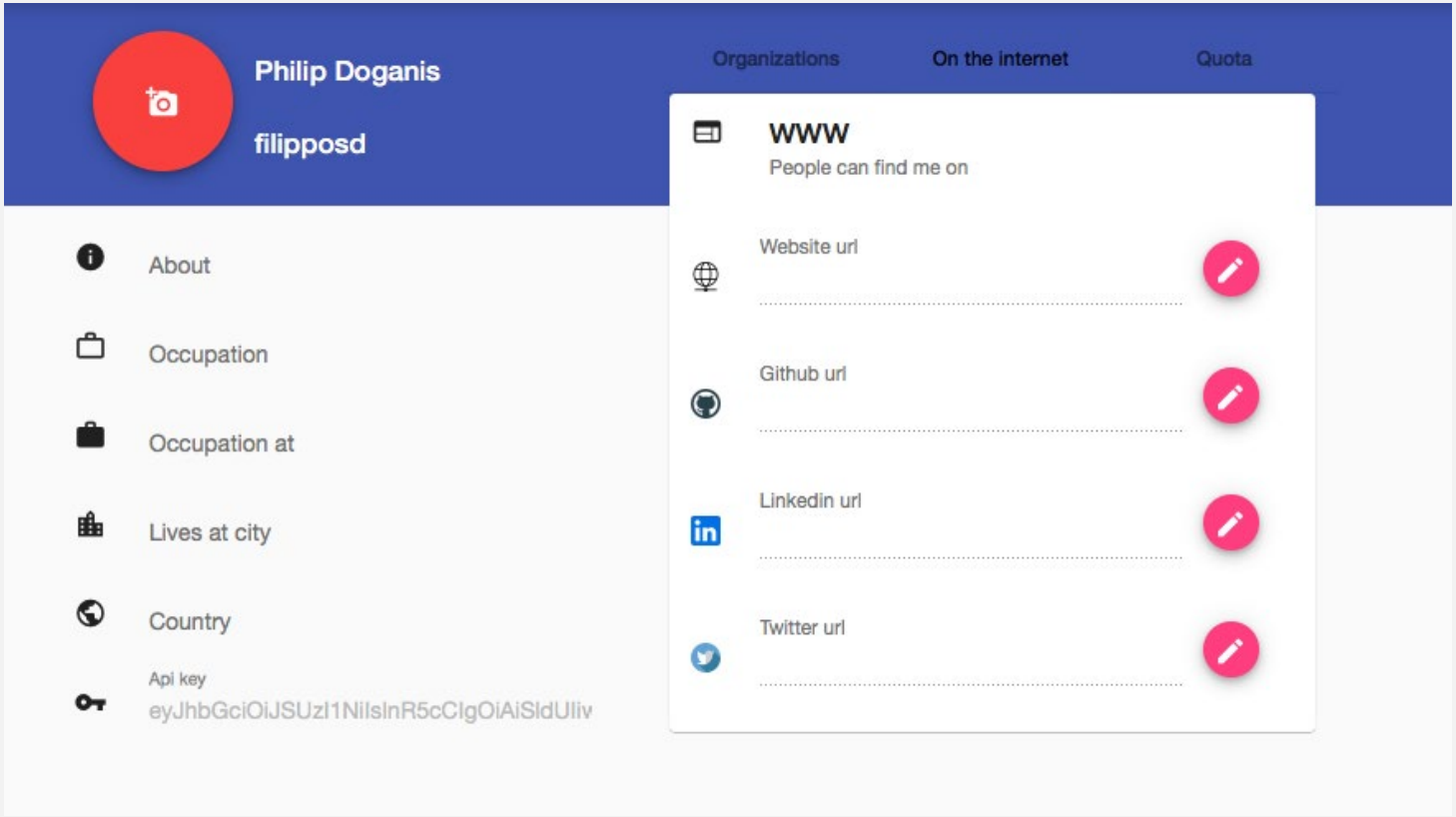
https://app.jaqpot.org/



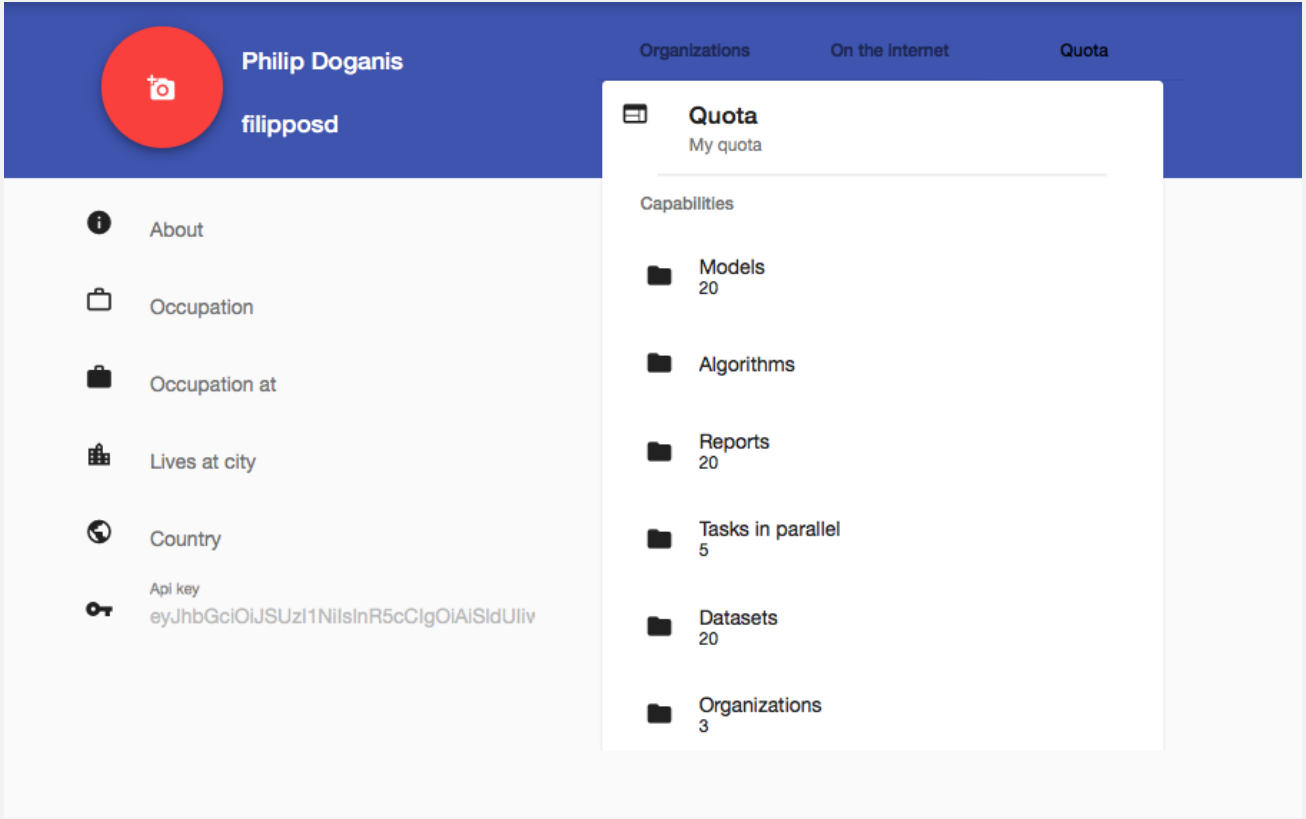
Jaqpot 5 - Login



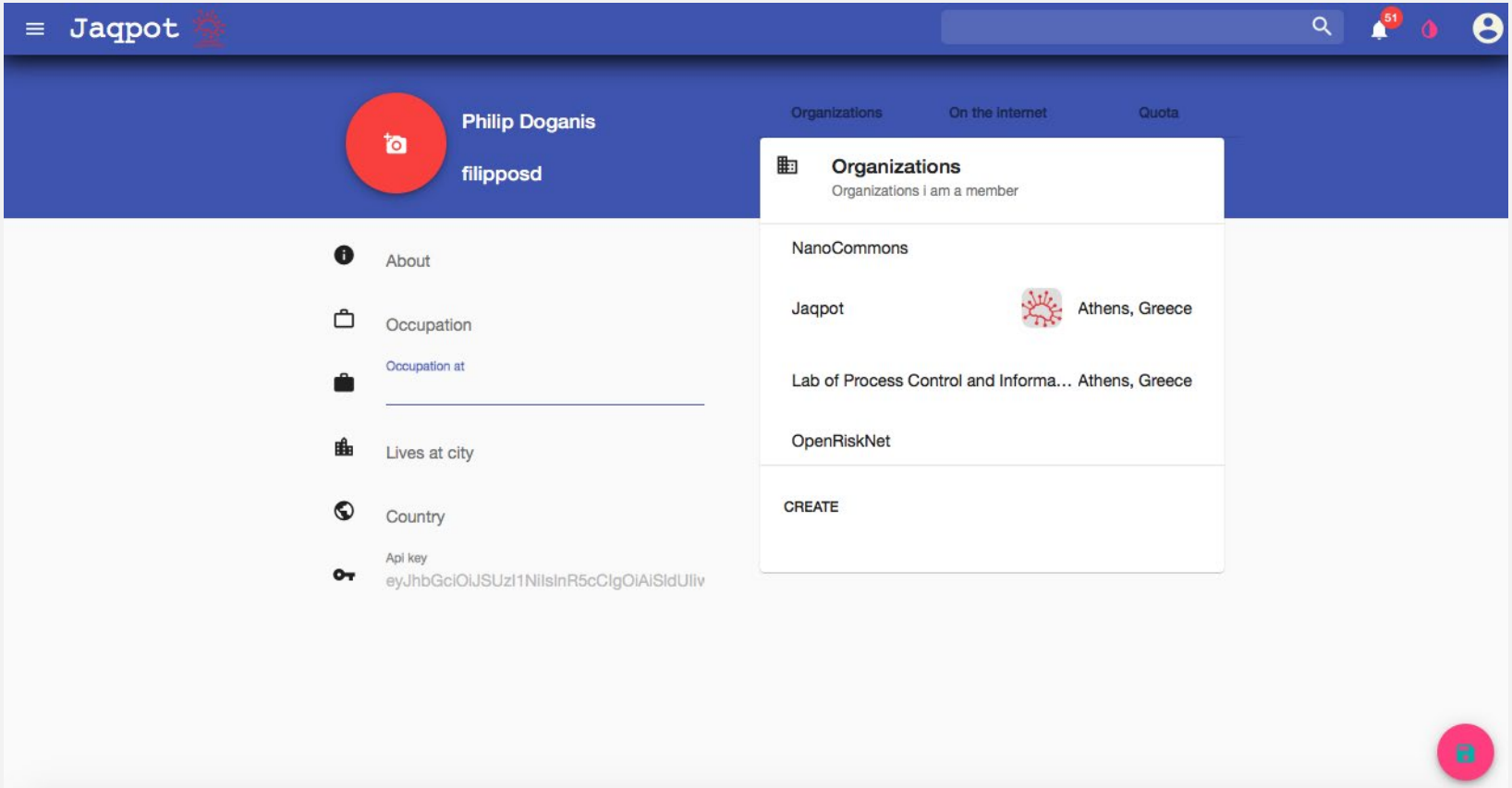
Jaqpot 5 - User profile - Contact Info



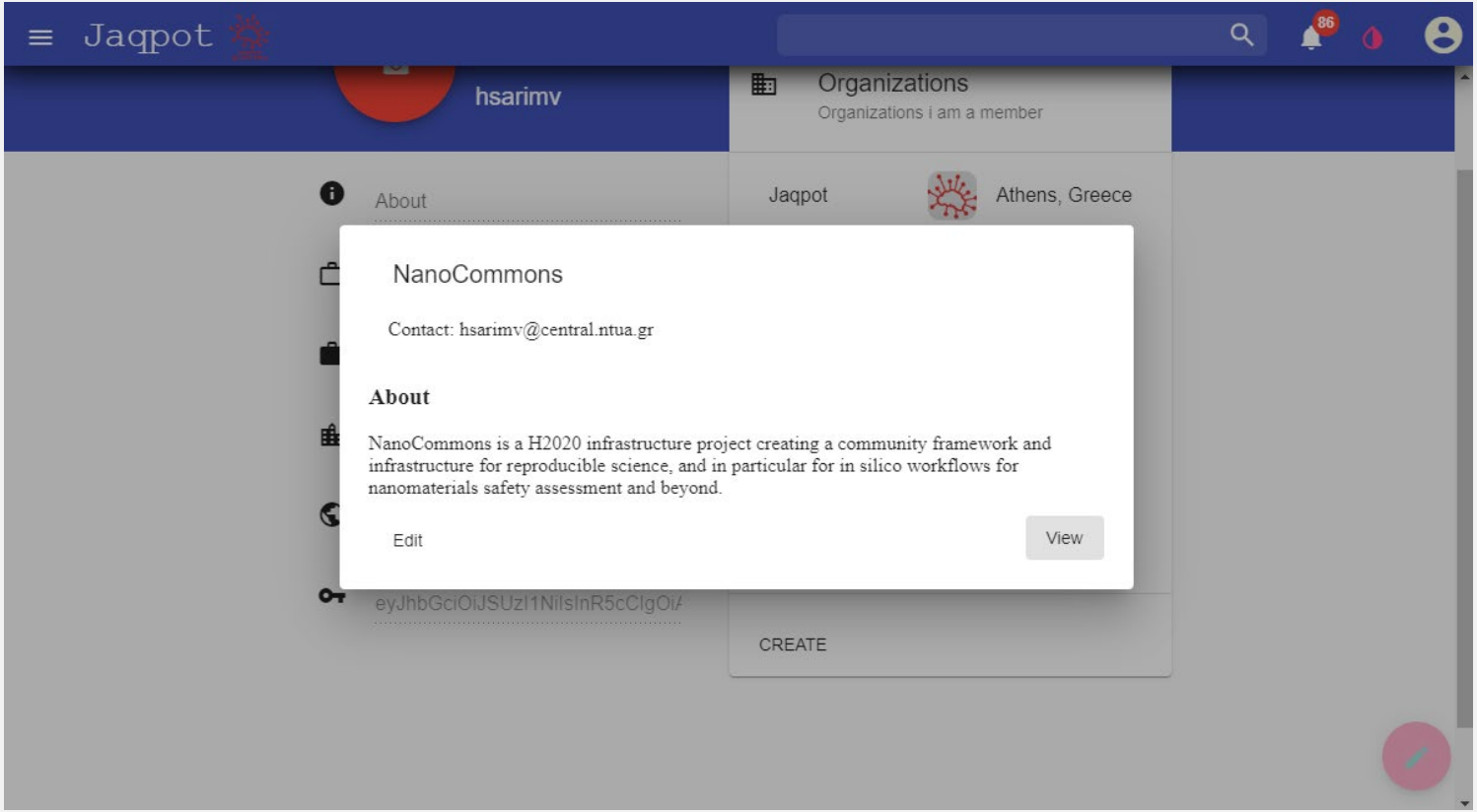
Jaqpot 5 - User profile - Quota



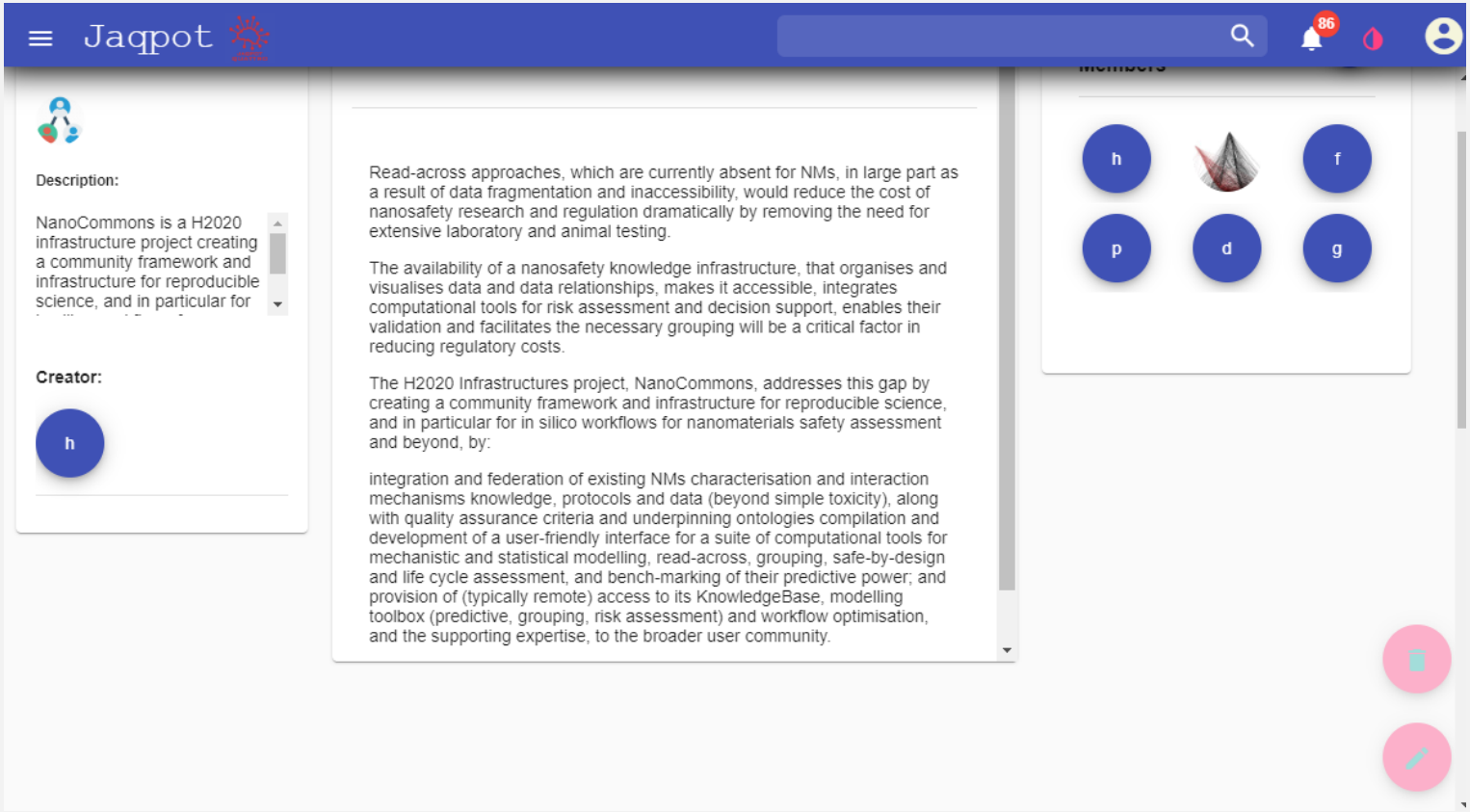
Jaqpot 5 - Organizations



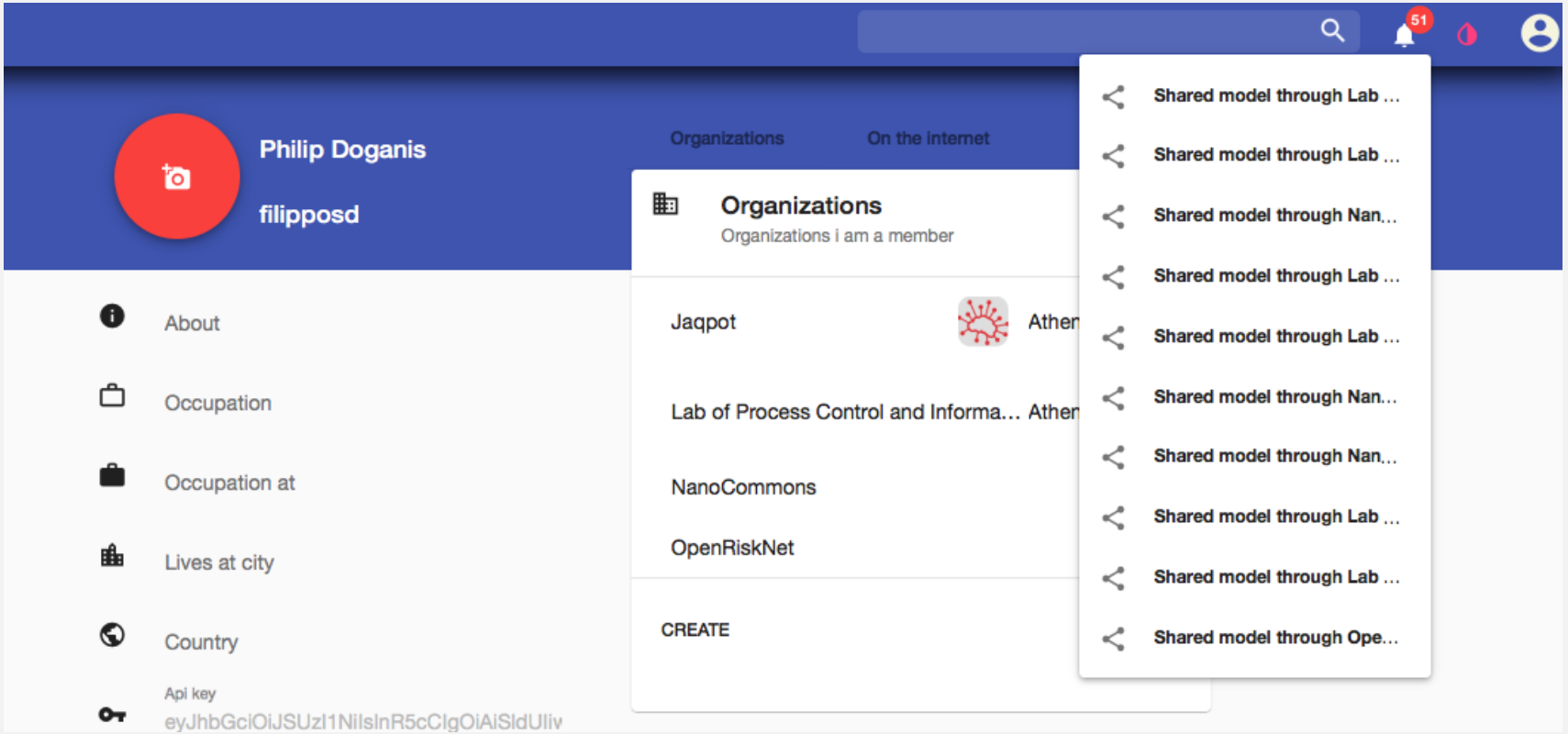
Jaqpot 5 - Organizations



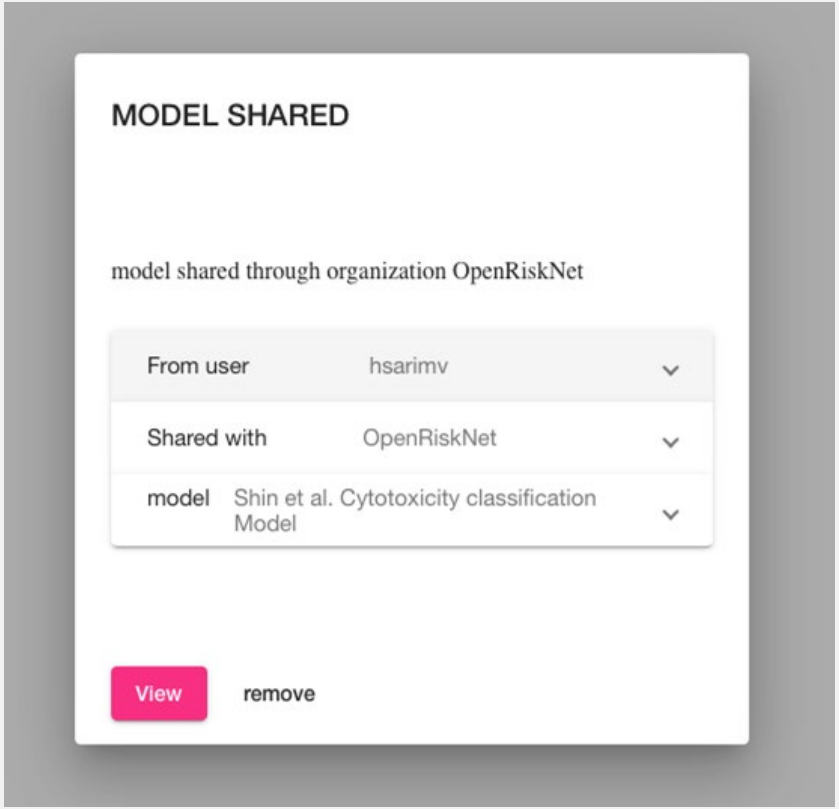
Jaqpot 5 - NanoCommons Organization page



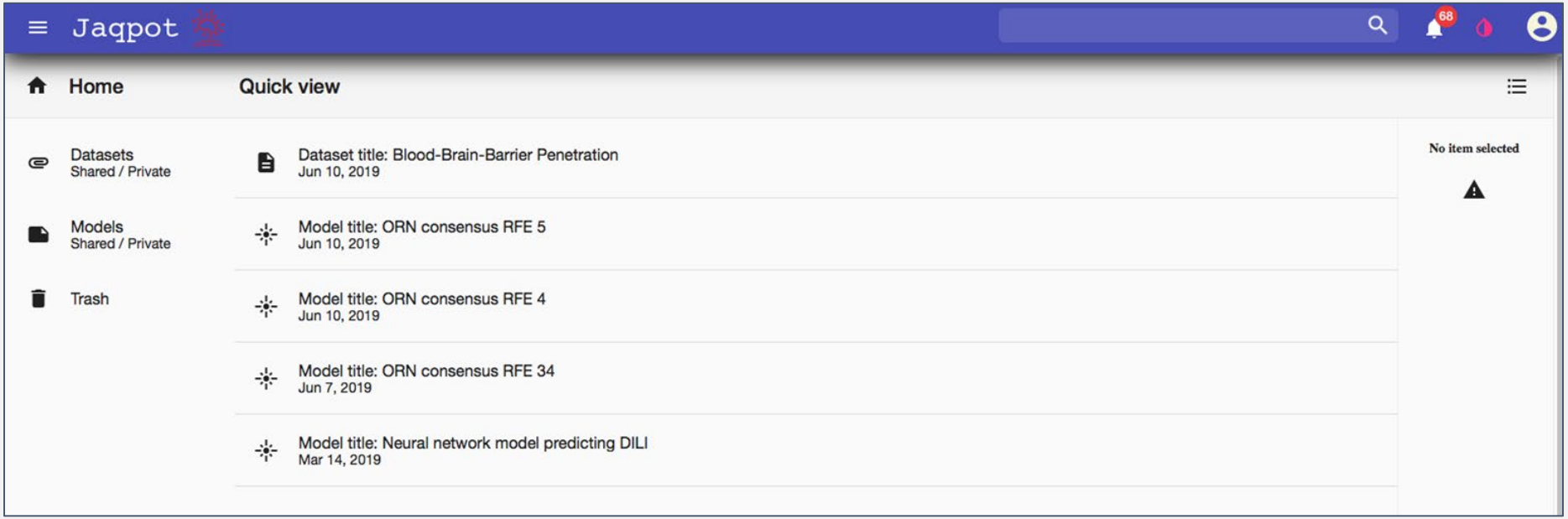
Jaqpot 5 - Sharing of resources



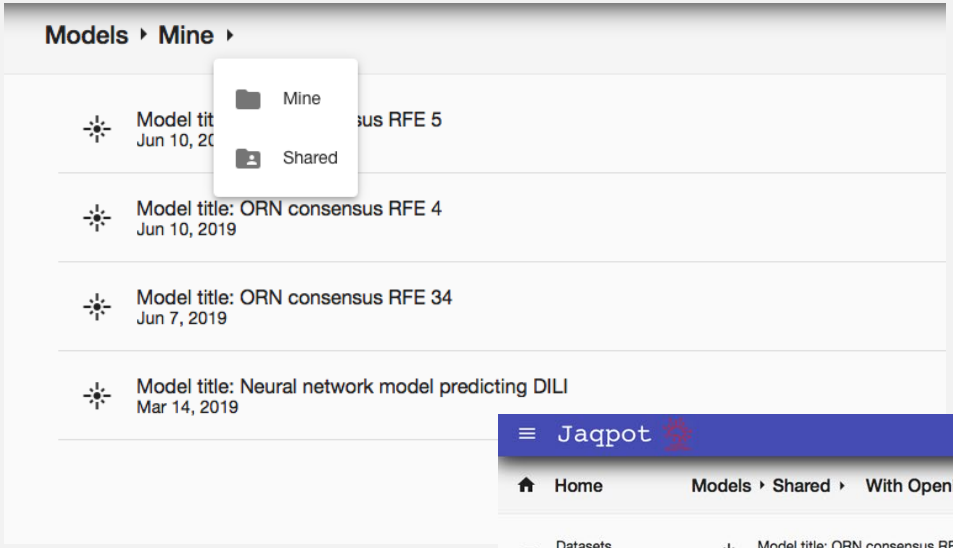
Jaqpot 5 - Sharing notification

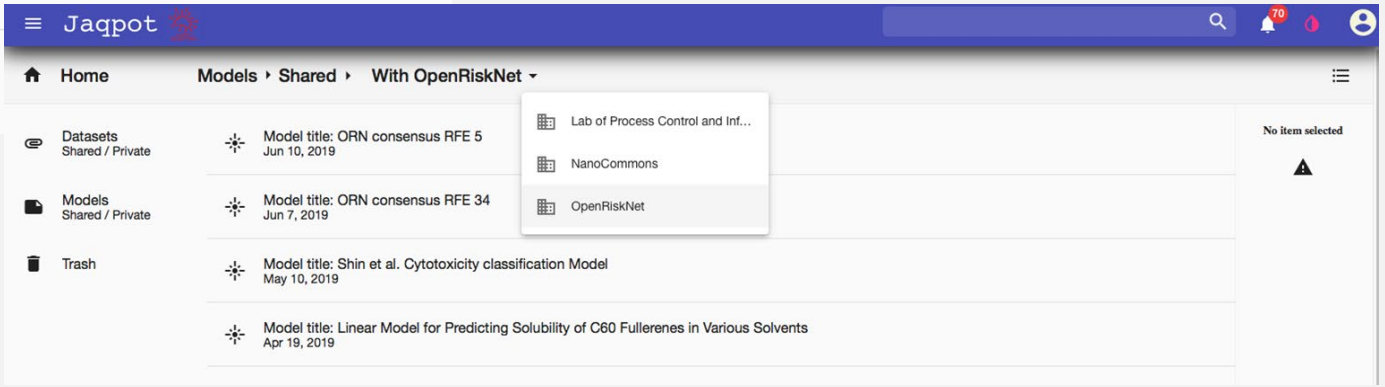


Jaqpot 5 - Home



Jaqpot 5 - Shared space - Models





The jaqpotpy library

https://jaqpotpy.readthedocs.io/en/latest/

- jaqpotpy is a python library that integrates fully the python **Scikit-learn** machine learning library with Jaqpot
- Only basic python programming is required to automatically deploy a predictive model as a web service in Jaqpot. Jupyter notebooks are developed to further assist model deployment.
- Similar libraries are under development for the integration of machine learning packages from other languages, R, Julia, Java.

```
from jaqpotpy import Jaqpot
import pandas as pd
from sklearn import linear_model




df2 = pd.read_csv('/path/train.csv')
X2 = df2[['Pclass',  'SibSp', 'Parch', 'Fare']]
y2 = df2['Survived']

clf = LogisticRegression(random_state=0, solver='lbfgs', multi_class='multinomial').fit(X2, y2)


jaqpot.deploy_linear_model(clf, X2, y2, title="Sklearn 2", description="Logistic regression model from python for the tita
                  algorithm="logistic regression")
```

Automated machine learning

https://jaqpotpy.readthedocs.io/en/latest/

Automated machine learning (**AutoML**) is the process of automating the end-to-end process of applying machine learning methodologies with the goal of maximizing the predictive performance of their final machine learning model:

- data pre-processing
- feature extraction
- feature selection
- algorithm selection
- hyperparameter optimization

Examples of AutoML platforms

- **DataRobots:** a commercial platform for AutoML
- **Auto-sklearn**: a Bayesian hyperparameter optimization layer on top of **scikit-learn**
- **TPOT**: a genetic progrmming optimization layer on top of **scikit-learn** that automatically creates and optimizes full machine learning pipelines.
- **Auto-WEKA**: a Bayesian hyperparameter optimization layer on top of **WEKA**
- **RRegrs**[1]: an exhaustive search layer on top of **Caret** (NTUA, University of Maastricht)

•[1]Tsiliki G, Munteanu CR, Seoane JA, Fernandez-Lozano C., Sarimveis H., Willighagen EL, (2015) RRegrs: an R package for computer-aided model selection with multiple regression models, Journal of Cheminformatics20157:46, https://doi.org/10.1186/s13321-015-0094-2

Jaqpot 5 – Create and deploy a nanoQSAR model using a Jupyter notebook

```
import pandas as pd
from jaqpotpy import Jaqpot
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import MinMaxScaler
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score, GridSearchCV, RandomizedSearchCV
df=pd.read_csv('C:\\MyDocuments\\European-funded\\Trexonta\\NanoCommons\\Actual Project\\WP5\\NanoQSAR deliverable\\
A Molecular Based Model for Prediction of Solubility of C60 in Various Solvents\\70_model_reduced.csv') # Reads the data
```

```
Xall=df[['piPC03', 'ATS1m', 'Seigp', 'More23e', 'H1m']] # Defines the columns that will be used as independent features
Yall=df['logS Exp.']   # Defines the end-point
X_train, X_test, Y_train, Y_test = train_test_split(Xall, Yall, train_size=0.75, test_size=0.25, random_state=1)
# Splits the data into training and test sets
stepslinear = [('scaler', MinMaxScaler()), ('MLR', LinearRegression())]
pipelinelinear = Pipeline(stepslinear) # define the pipeline object.
cross_val_score(estimator=pipelinelinear, X=X_train, y=Y_train, cv=5, n_jobs=-1)  #Performs a 5-fold cross validation
```

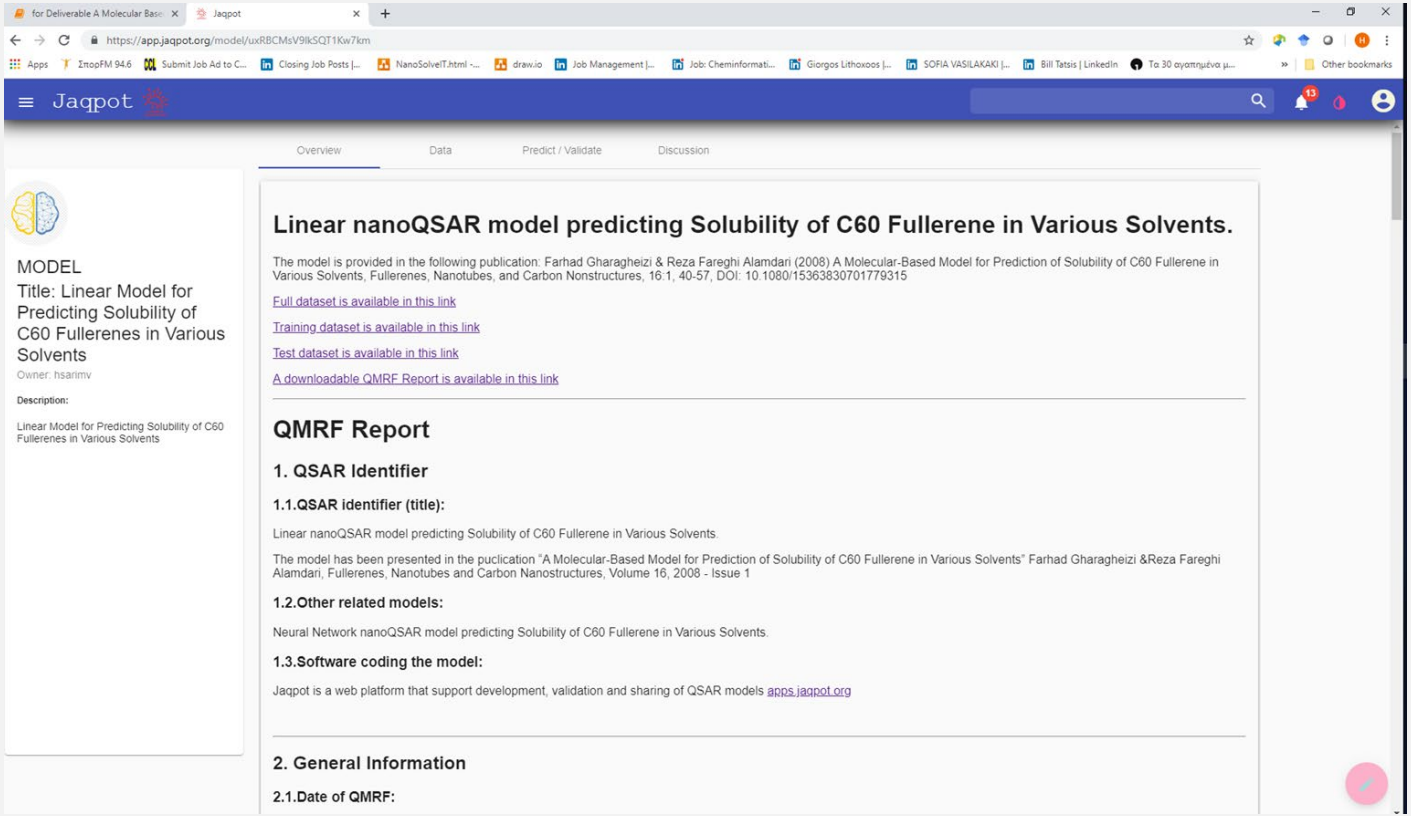
```
pipelinelinear.fit(X_train, Y_train)
print('Training score: ', pipelinelinear.score(X_train, Y_train))
print('Testing score: ', pipelinelinear.score(X_test, Y_test))
print('Total score: ', pipelinelinear.score(Xall, Yall))              #Trains the model and prints R^2 statistics
```

```
jaqpot = Jaqpot("https://api.jaqpot.org/jaqpot/services/")
jaqpot.request_key_safe()
```

```
jaqpot.deploy_pipeline(pipelinelinear,Xall,Yall,"Linear Model for Predicting Solubility of C60 Fullerenes in Various Solvents","Linear
Model","linearmodel")
```


Jaqpot 5 - Model page

https://app.jaqpot.org/model/uxRBCMsV9lkSQT1Kw7km



Jaqpot 5 - QMRF report

QMRF identifier (JRC Inventory): **To be entered by JRC**

QMRF Title: **Linear nanoQSAR model predicting Solubility of C60 Fullerene in Various Solvents. The model has been presented in the puclication "A Molecular?Based Model for Prediction of Solubility of C60 Fullerene in Various Solvents" Farhad Gharagheizi &Reza Fareghi Alamdari, Fullerenes, Nanotubes and Carbon Nanostructures, Volume 16, 2008 - Issue 1**

Printing Date: **22-Apr-2019**

1.QSAR identifier

1.1.QSAR identifier (title):

Linear nanoQSAR model predicting Solubility of C60 Fullerene in Various Solvents. The model has been presented in the puclication "A Molecular?Based Model for Prediction of Solubility of C60 Fullerene in Various Solvents" Farhad Gharagheizi &Reza Fareghi Alamdari, Fullerenes, Nanotubes and Carbon Nanostructures, Volume 16, 2008 - Issue 1

1.2.Other related models:

Neural Network nanoQSAR model predicting Solubility of C60 Fullerene in Various Solvents.

1.3.Software coding the model:

Jaqpot

Jaqpot is q web platform that support development, validation and sharing of QSAR models

Haralambos Sarimveis

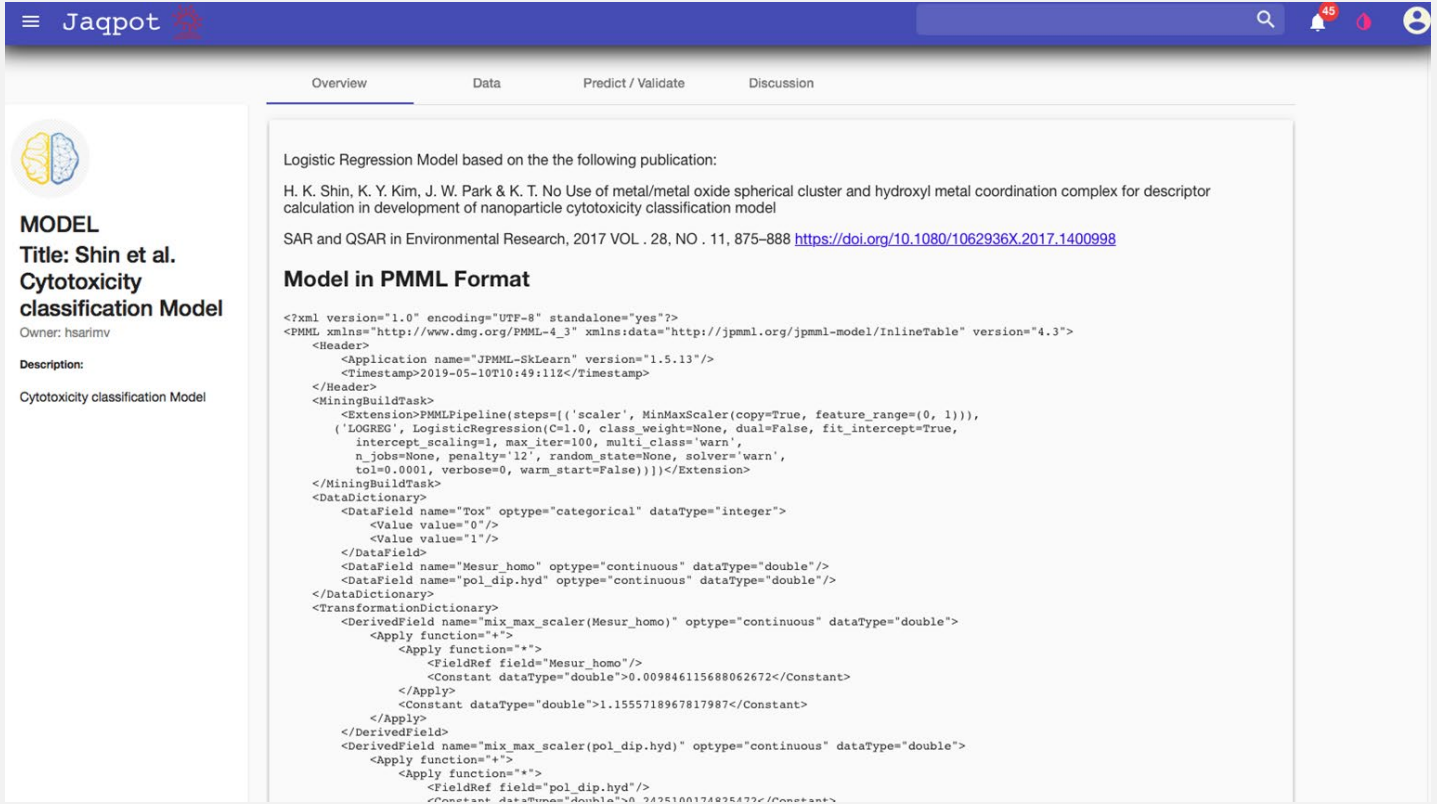
apps.jaqpot.org

2.General information

2.1.Date of QMRF:

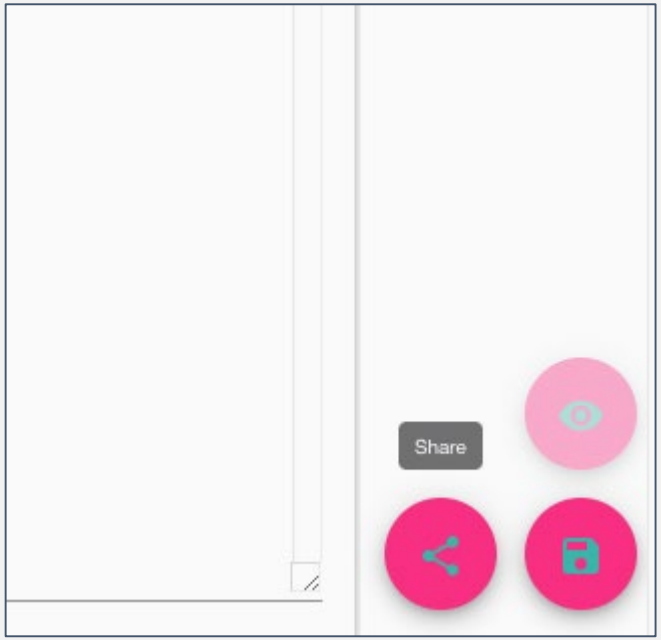
21 April 2019

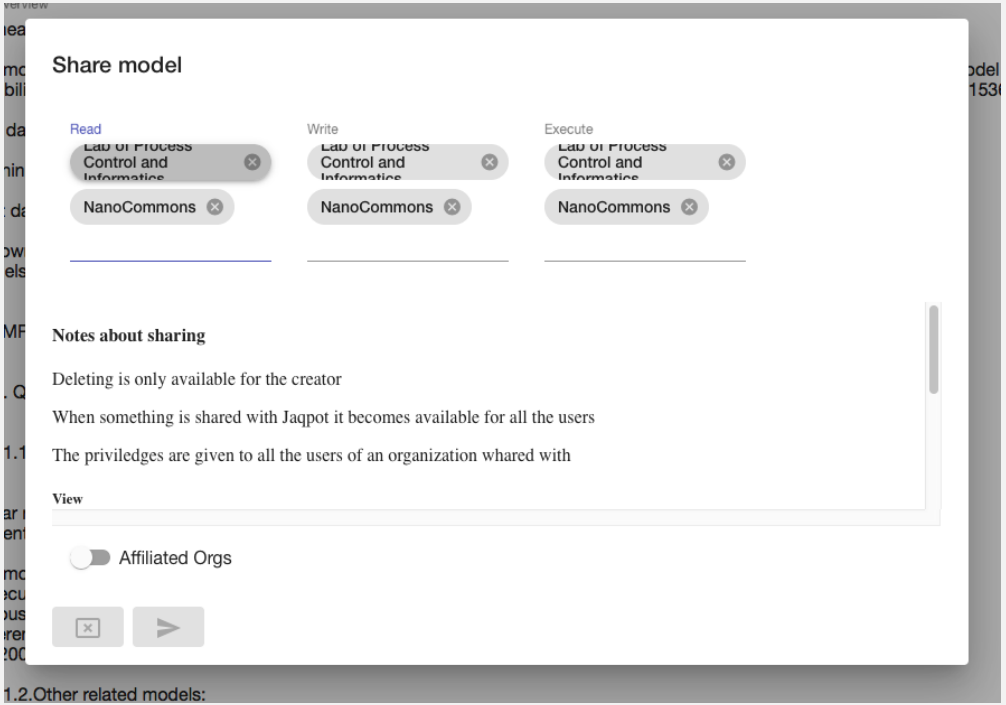
Jaqpot 5 – PMML representation



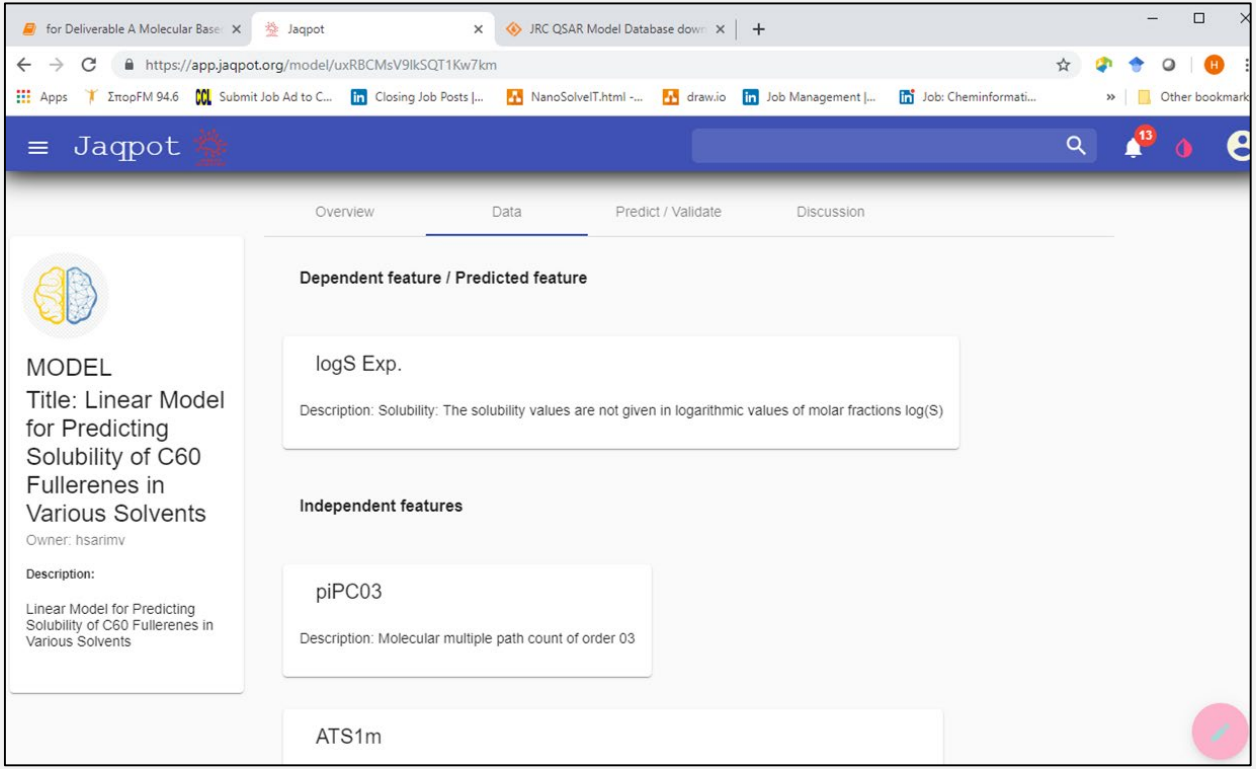
https://app.jaqpot.org/model/uxRBCMsV9IkSQTIKw7km

Jaqpot 5 - Model page

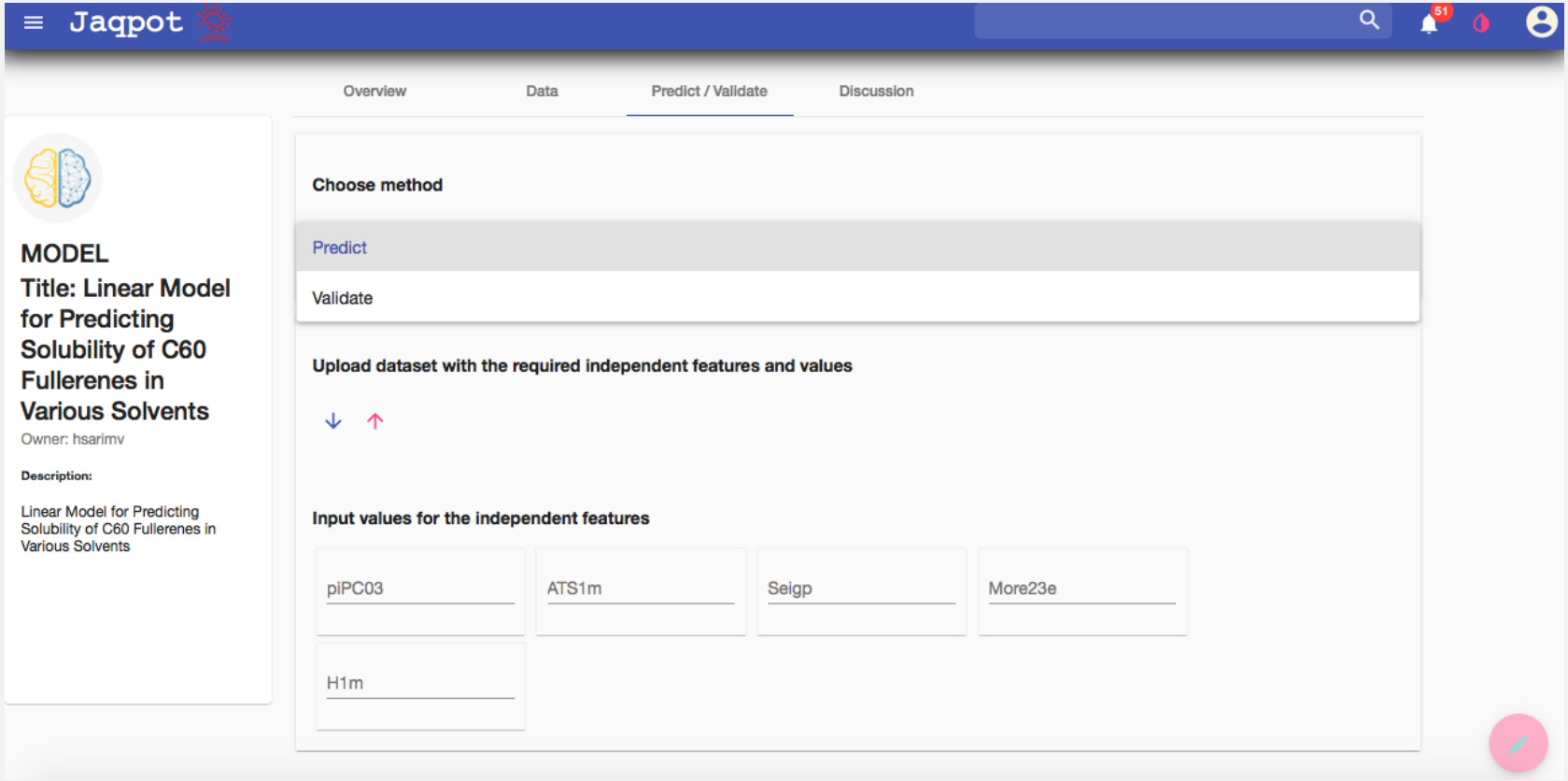




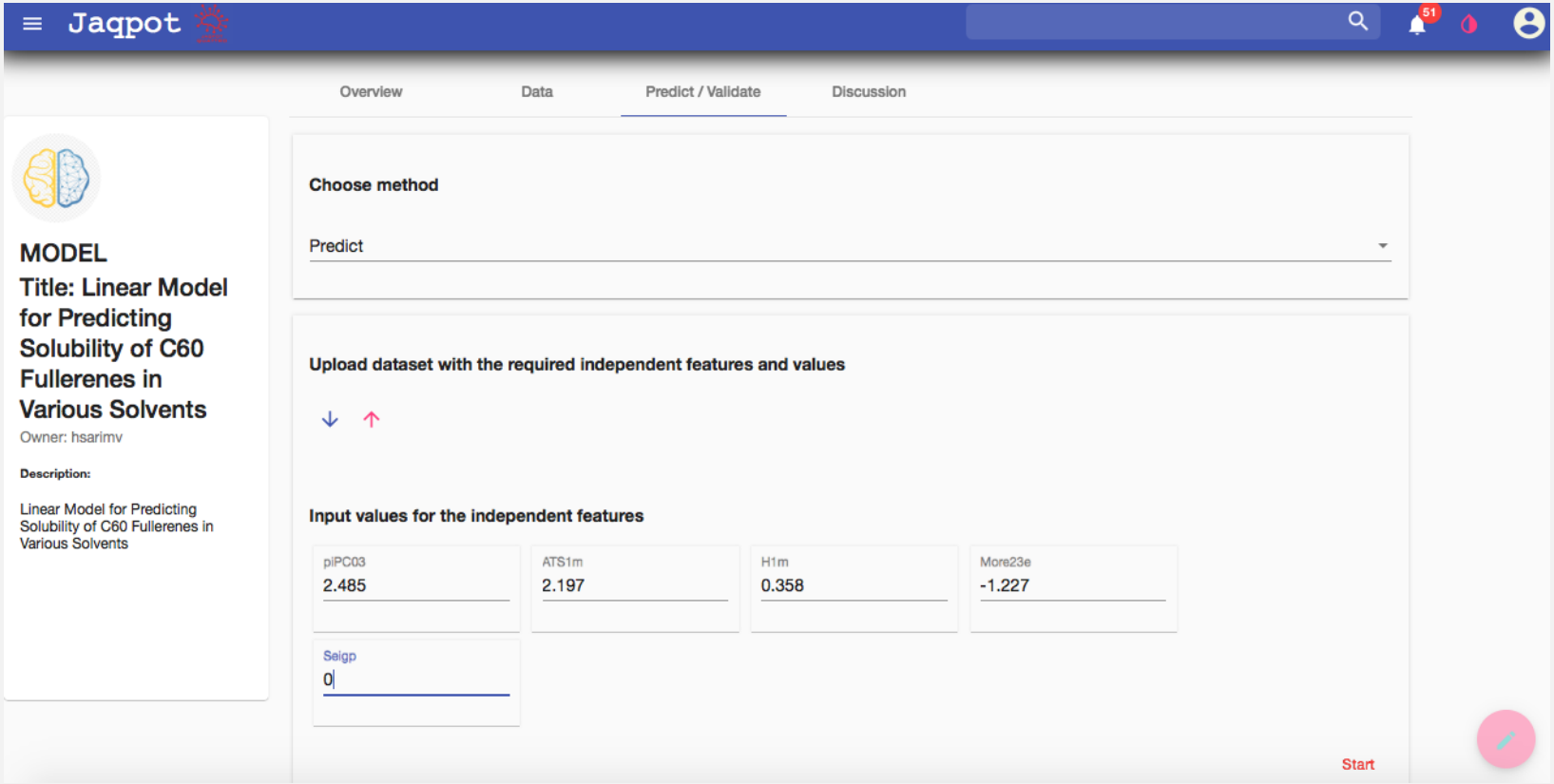
Jaqpot 5 - Model page - Data tab



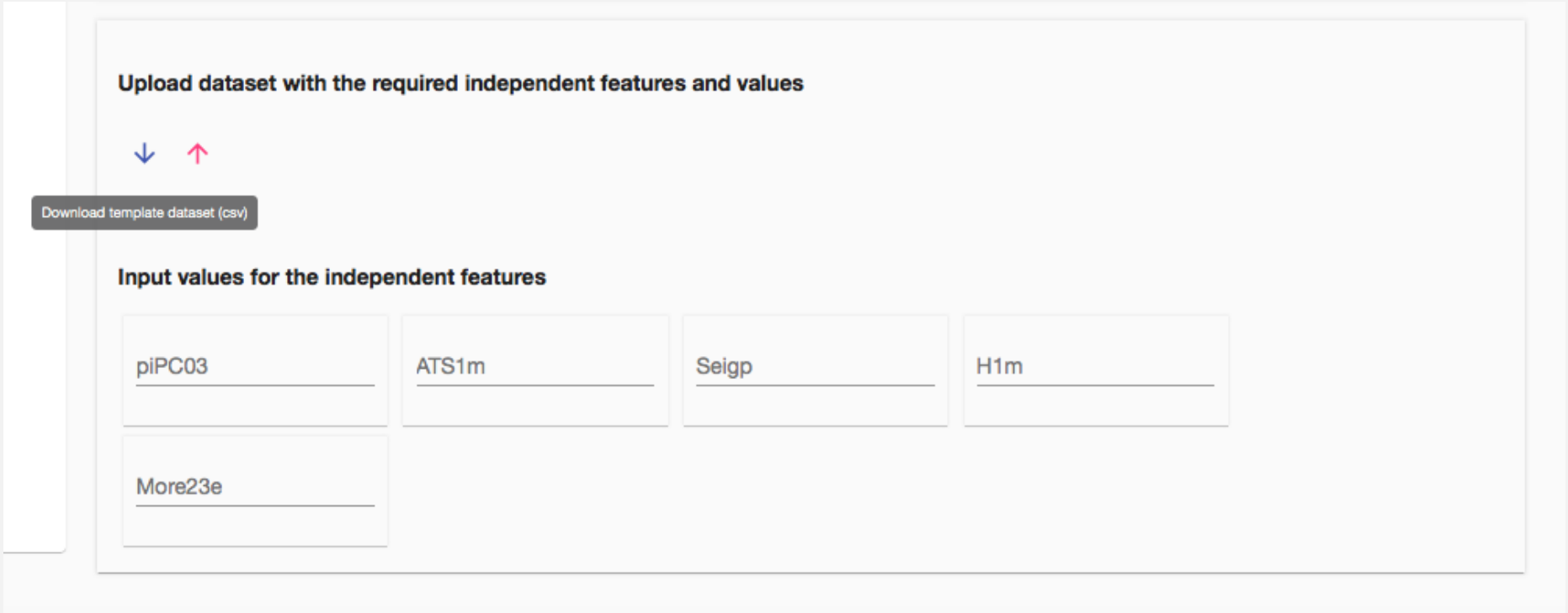
Jaqpot 5 - Model page – Predict/Validate tab



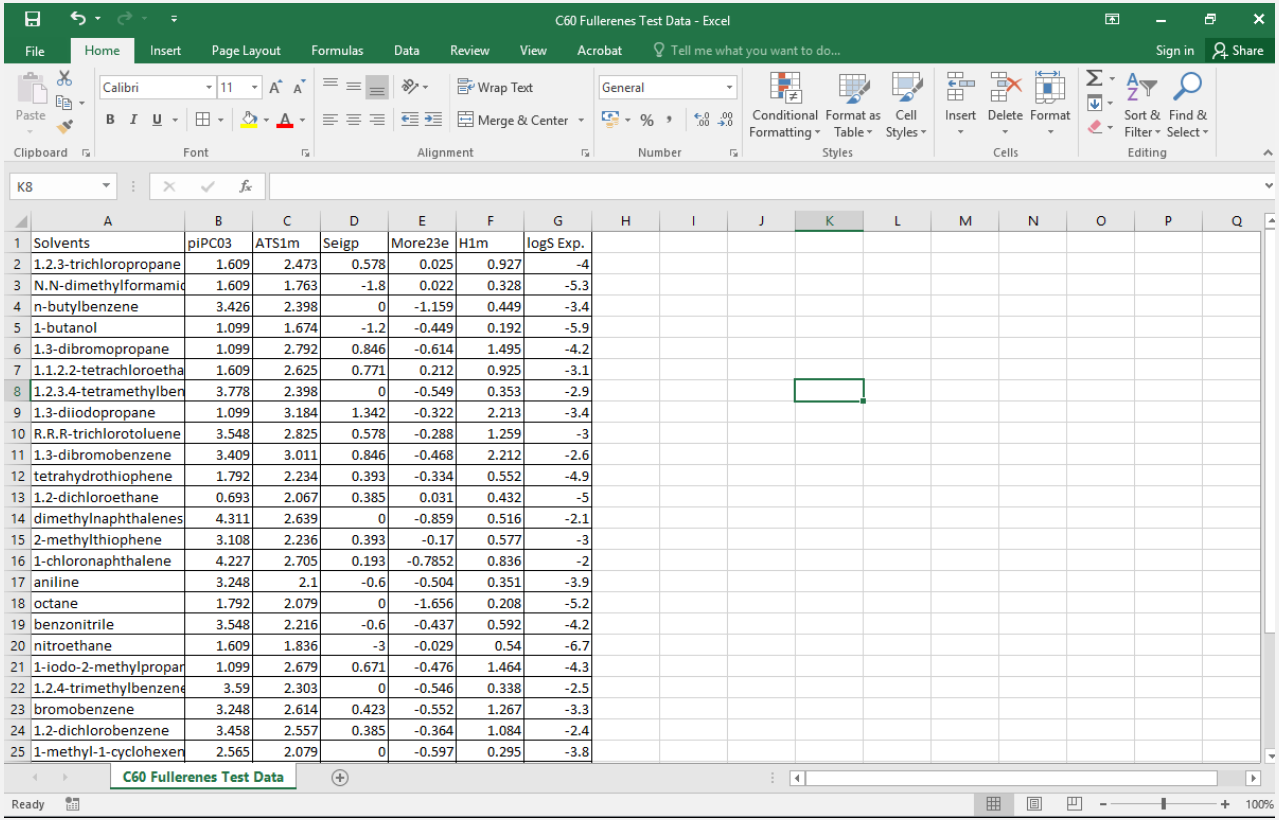
Jaqpot 5 - Predict - Entering values



Jaqpot 5 - Predict - Downloading template

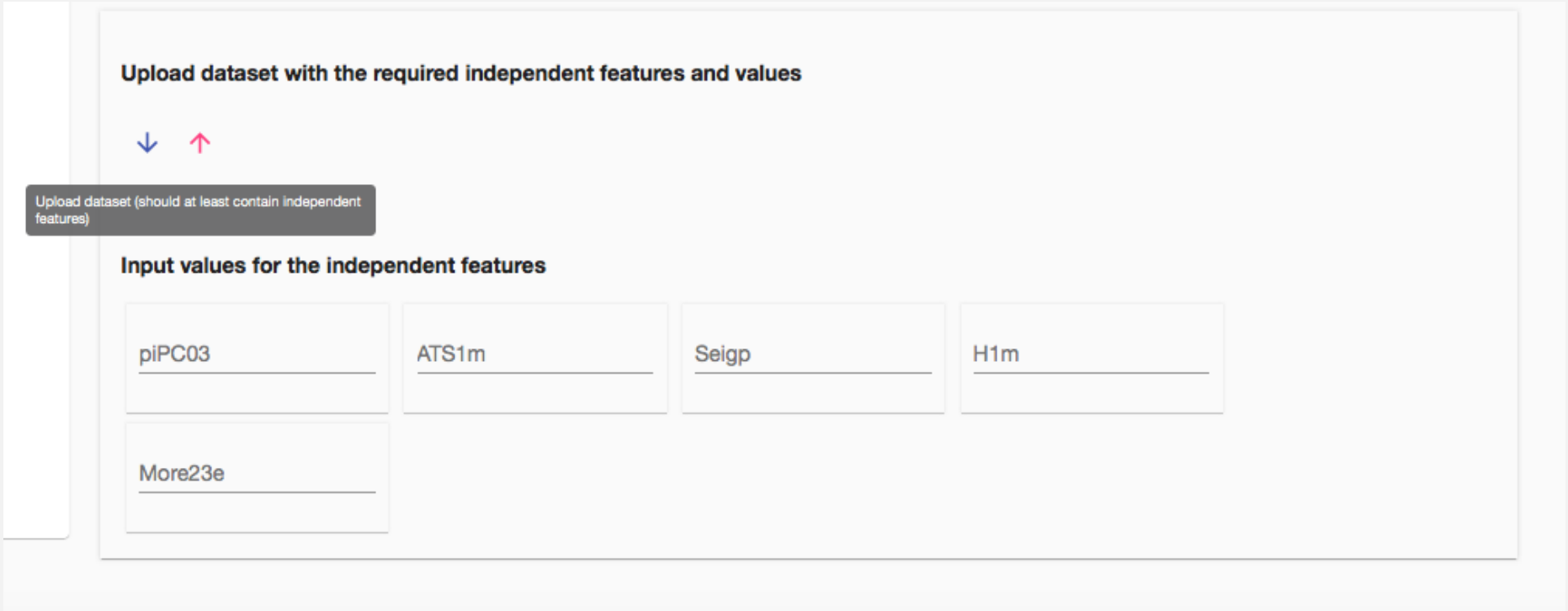


Jaqpot 5 - Predict - Editing template

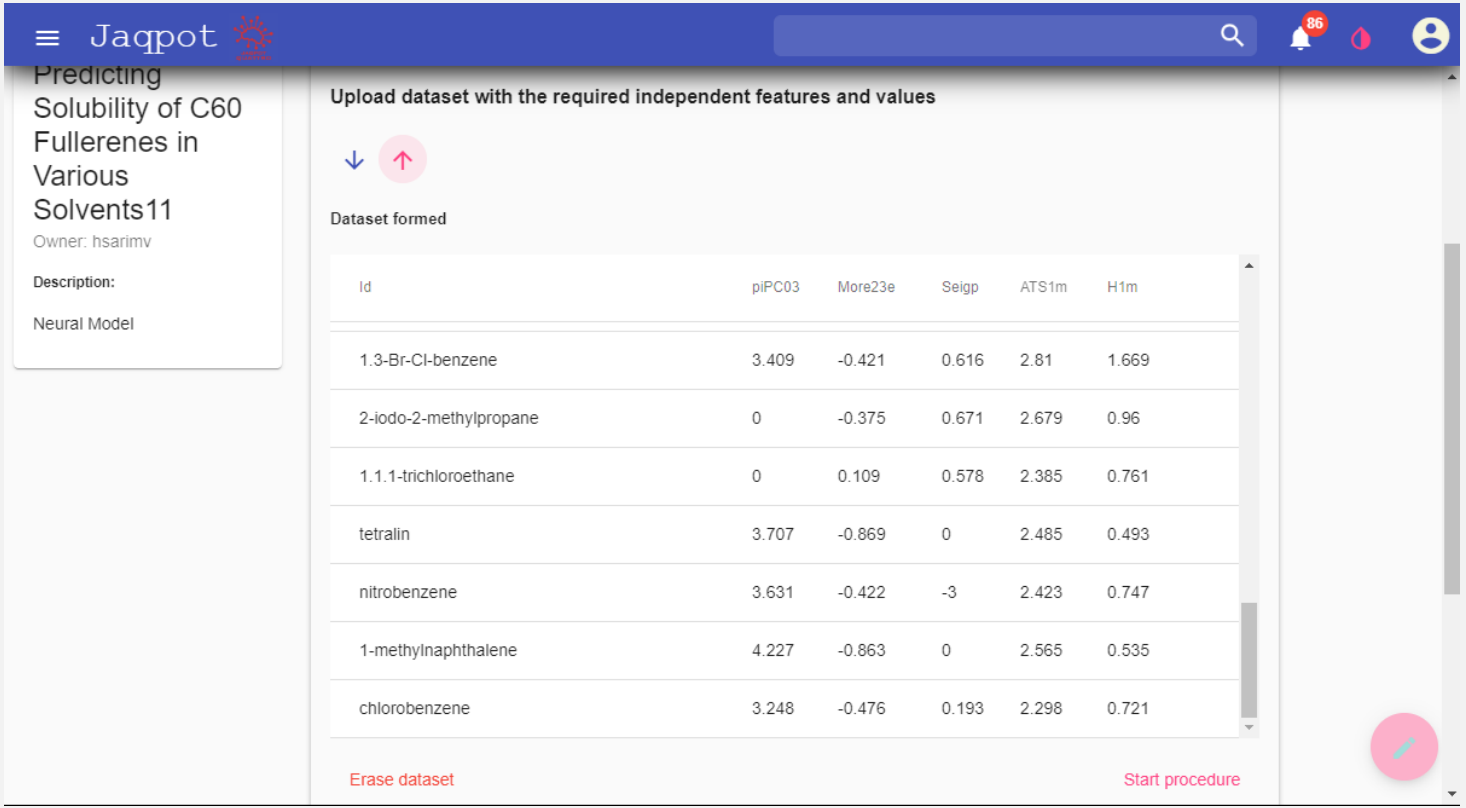


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Solvents	piPC03	ATS1m	Seigp	More23e	H1m	logS Exp.										
2	1.2.3-trichloropropane	1.609	2.473	0.578	0.025	0.927	-4										
3	N.N-dimethylformamid	1.609	1.763	-1.8	0.022	0.328	-5.3										
4	n-butylbenzene	3.426	2.398	0	-1.159	0.449	-3.4										
5	1-butanol	1.099	1.674	-1.2	-0.449	0.192	-5.9										
6	1.3-dibromopropane	1.099	2.792	0.846	-0.614	1.495	-4.2										
7	1.1.2.2-tetrachloroetha	1.609	2.625	0.771	0.212	0.925	-3.1										
8	1.2.3.4-tetramethylben	3.778	2.398	0	-0.549	0.353	-2.9										
9	1.3-diiodopropane	1.099	3.184	1.342	-0.322	2.213	-3.4										
10	R.R.R-trichlorotoluene	3.548	2.825	0.578	-0.288	1.259	-3										
11	1.3-dibromobenzene	3.409	3.011	0.846	-0.468	2.212	-2.6										
12	tetrahydrothiophene	1.792	2.234	0.393	-0.334	0.552	-4.9										
13	1.2-dichloroethane	0.693	2.067	0.385	0.031	0.432	-5										
14	dimethylnaphthalenes	4.311	2.639	0	-0.859	0.516	-2.1										
15	2-methylthiophene	3.108	2.236	0.393	-0.17	0.577	-3										
16	1-chloronaphthalene	4.227	2.705	0.193	-0.7852	0.836	-2										
17	aniline	3.248	2.1	-0.6	-0.504	0.351	-3.9										
18	octane	1.792	2.079	0	-1.656	0.208	-5.2										
19	benzonitrile	3.548	2.216	-0.6	-0.437	0.592	-4.2										
20	nitroethane	1.609	1.836	-3	-0.029	0.54	-6.7										
21	1-iodo-2-methylpropan	1.099	2.679	0.671	-0.476	1.464	-4.3										
22	1.2.4-trimethylbenzene	3.59	2.303	0	-0.546	0.338	-2.5										
23	bromobenzene	3.248	2.614	0.423	-0.552	1.267	-3.3										
24	1.2-dichlorobenzene	3.458	2.557	0.385	-0.364	1.084	-2.4										
25	1-methyl-1-cyclohexen	2.565	2.079	0	-0.597	0.295	-3.8										

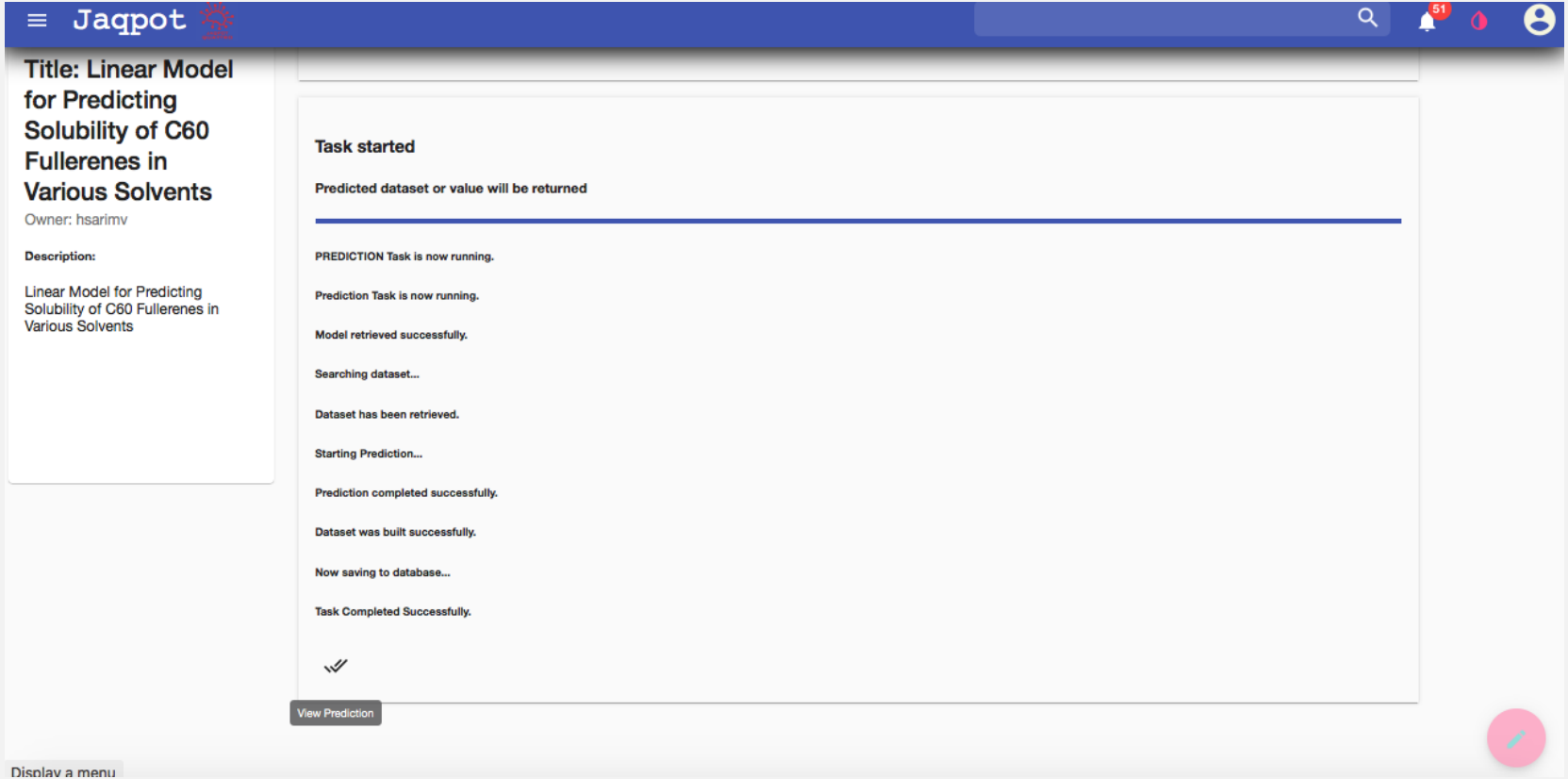
Jaqpot 5 - Predict - Uploading data



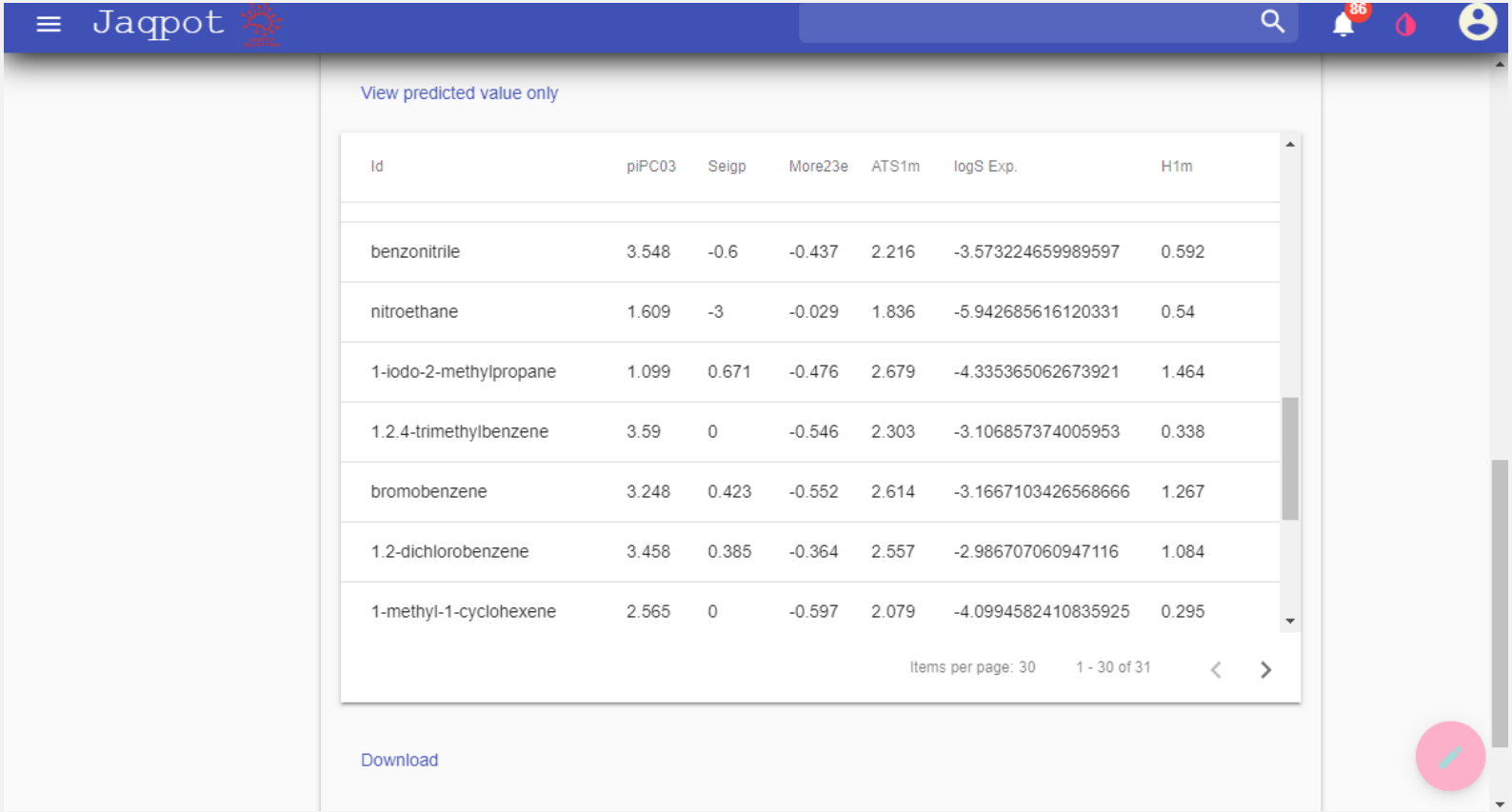
Jaqpot 5 - Predict - Uploading data



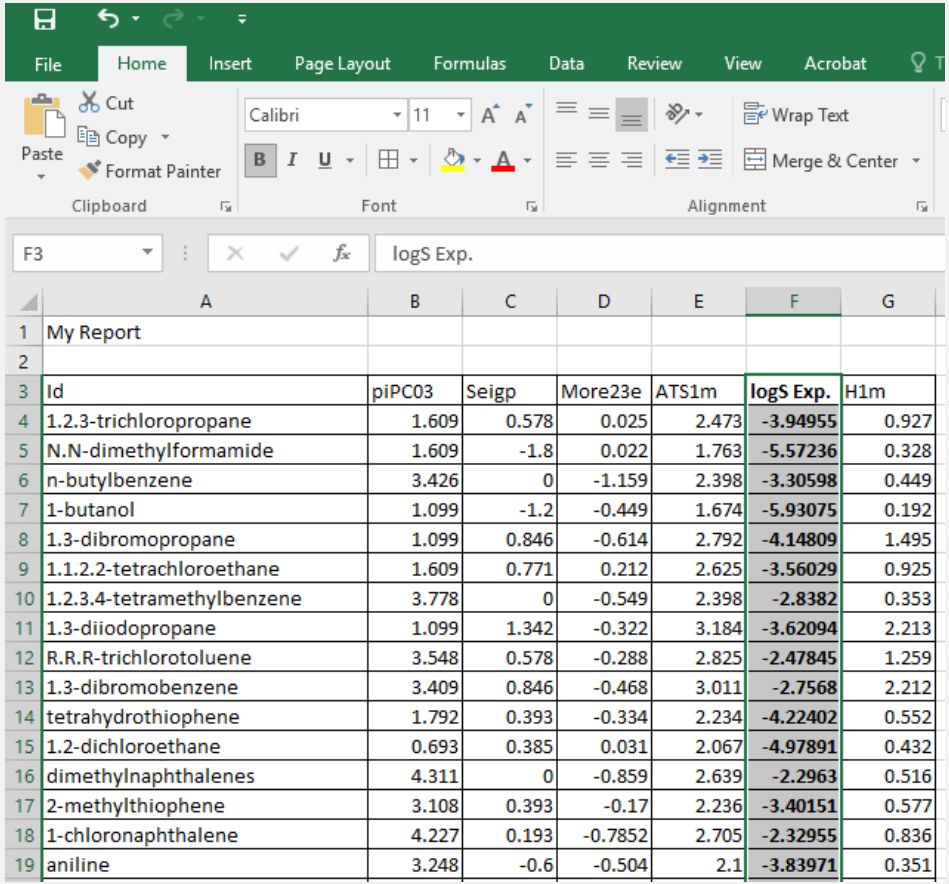
Jaqpot 5 – Calculating predictions



Jaqpot 5 – Displaying predictions

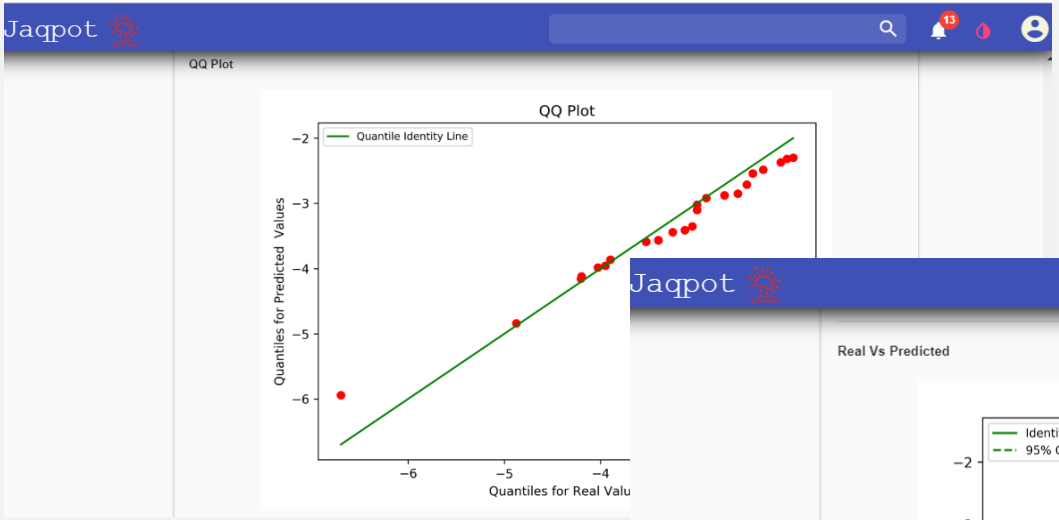


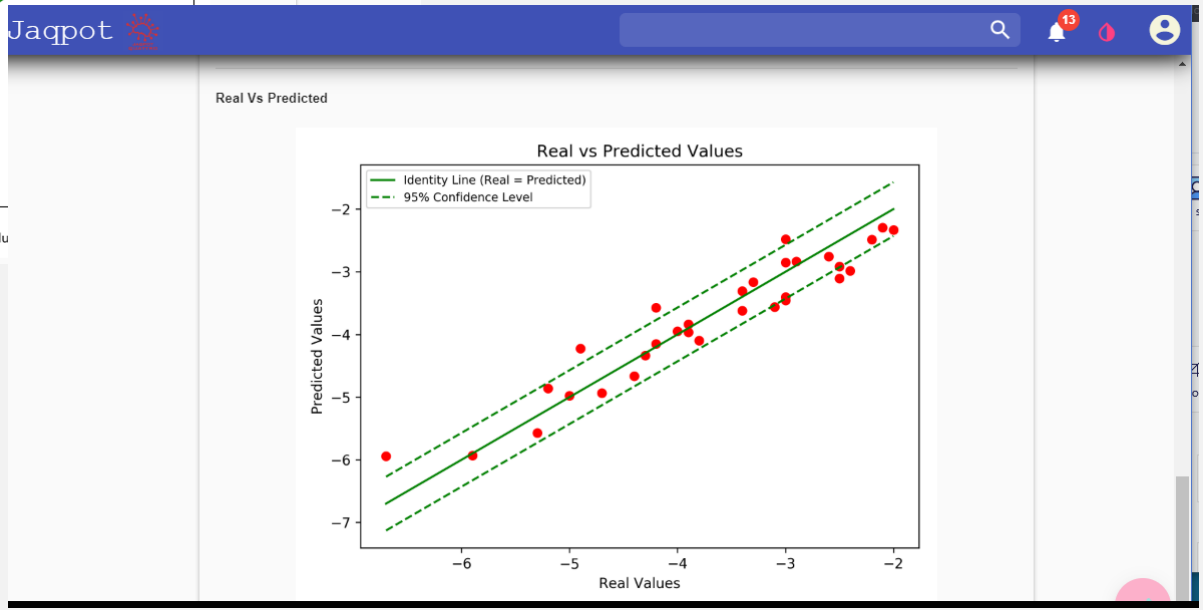
Jaqpot 5 - Predictions as CSV



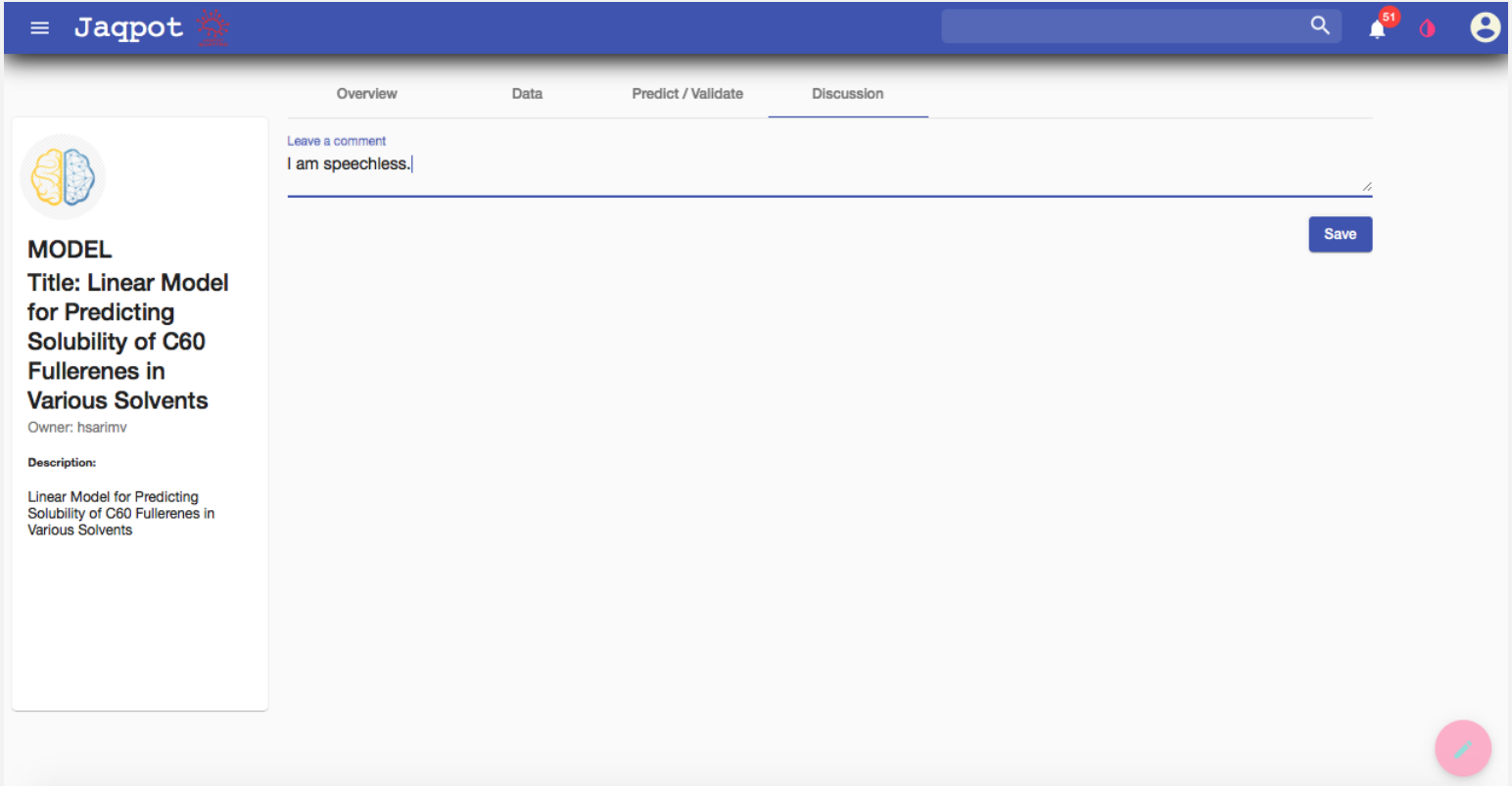
	A	B	C	D	E	F	G
1	My Report						
2							
3	Id	piPC03	Seigp	More23e	ATS1m	logS Exp.	H1m
4	1.2.3-trichloropropane	1.609	0.578	0.025	2.473	-3.94955	0.927
5	N.N-dimethylformamide	1.609	-1.8	0.022	1.763	-5.57236	0.328
6	n-butylbenzene	3.426	0	-1.159	2.398	-3.30598	0.449
7	1-butanol	1.099	-1.2	-0.449	1.674	-5.93075	0.192
8	1.3-dibromopropane	1.099	0.846	-0.614	2.792	-4.14809	1.495
9	1.1.2.2-tetrachloroethane	1.609	0.771	0.212	2.625	-3.56029	0.925
10	1.2.3.4-tetramethylbenzene	3.778	0	-0.549	2.398	-2.8382	0.353
11	1.3-diiodopropane	1.099	1.342	-0.322	3.184	-3.62094	2.213
12	R.R.R-trichlorotoluene	3.548	0.578	-0.288	2.825	-2.47845	1.259
13	1.3-dibromobenzene	3.409	0.846	-0.468	3.011	-2.7568	2.212
14	tetrahydrothiophene	1.792	0.393	-0.334	2.234	-4.22402	0.552
15	1.2-dichloroethane	0.693	0.385	0.031	2.067	-4.97891	0.432
16	dimethylnaphthalenes	4.311	0	-0.859	2.639	-2.2963	0.516
17	2-methylthiophene	3.108	0.393	-0.17	2.236	-3.40151	0.577
18	1-chloronaphthalene	4.227	0.193	-0.7852	2.705	-2.32955	0.836
19	aniline	3.248	-0.6	-0.504	2.1	-3.83971	0.351

Jaqpot 5 – Model validation





Jaqpot 5 - Social network of models

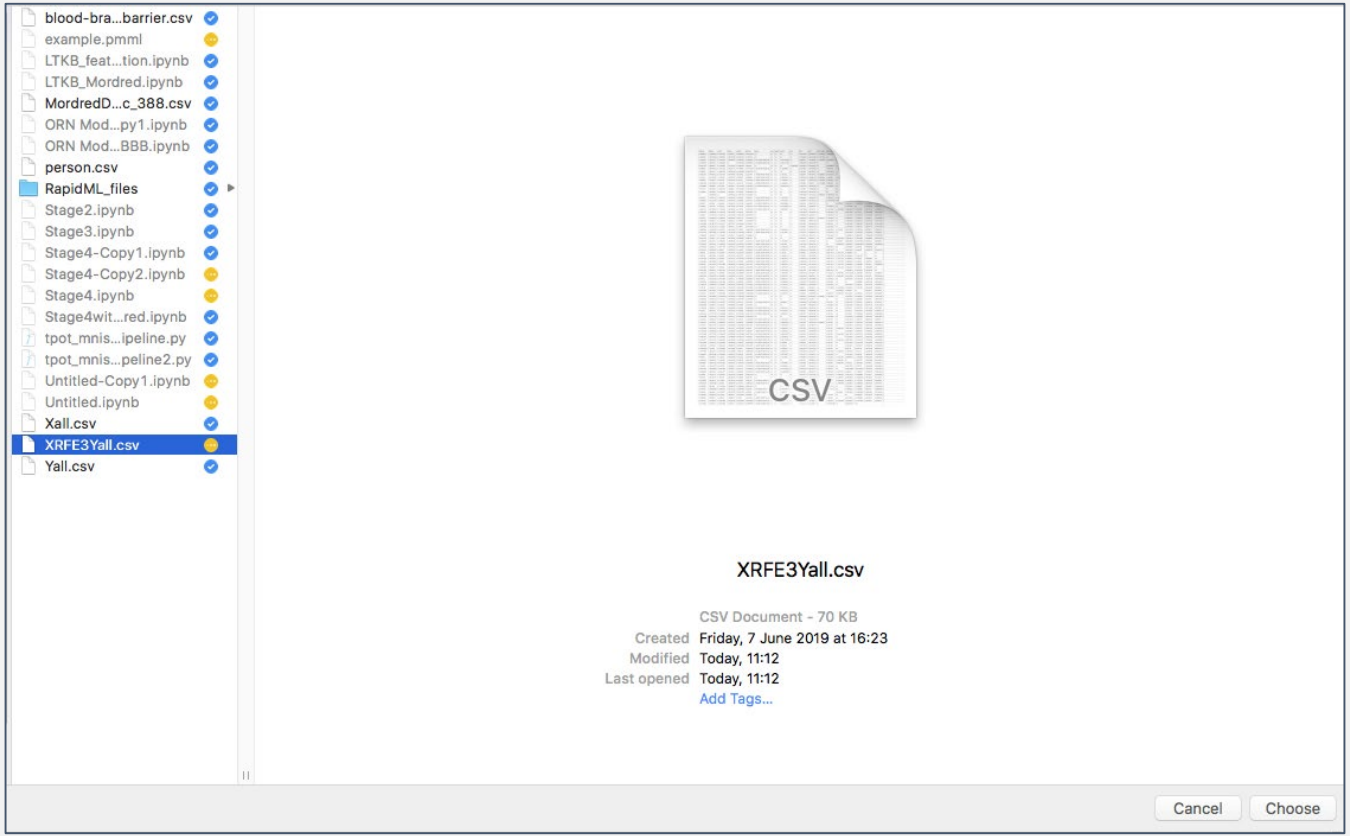


Jaqpot library of nanoQSAR models

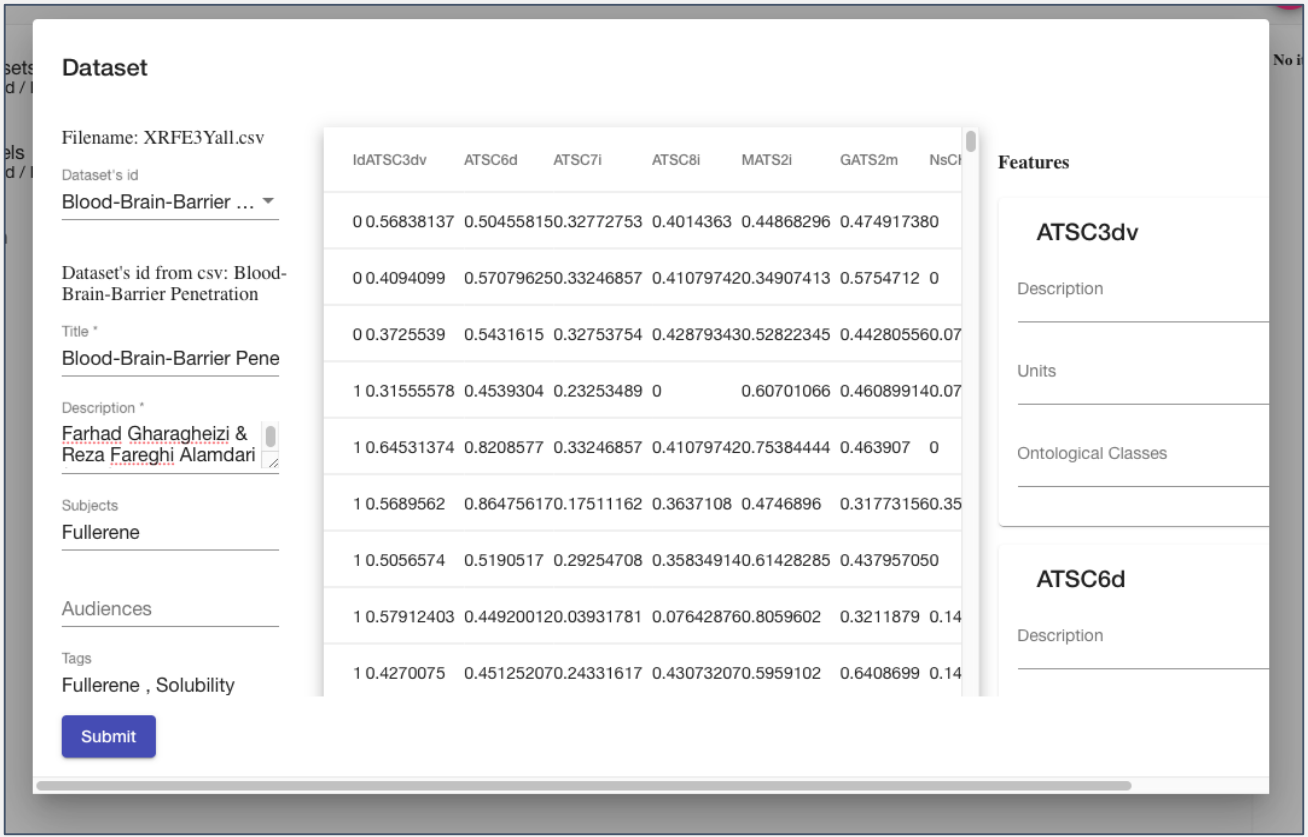
Jaqpot 4/NanoCommons account, Jaqpot 5/ Nanocommons organization

	Model description	Endpoint	Model URIs
1	Methodology for developing structure-activity evaluation to identify combinations of physical features of nanomaterial that influence potential cell damage by MLR/LDA (TiO2 case) **Sayes & Ivanov (2010)**	In vitro - Cytotoxicity - membrane damage measured as lactate dehydrogenase (LDH) release [units/L]	Jaqpot 4: http://jaqpot.org/m_detail?name=2KoKHcIgMJloSeWuZ03a Jaqpot 5 https://app.jaqpot.org/model/izMncmc5LMbgC6o7Fkj8
2	Methodology for developing structure-activity evaluation to identify combinations of physical features of nanomaterial that influence potential cell damage by MLR/LDA (ZnO case) **Sayes & Ivanov (2010)**	In vitro - Cytotoxicity - membrane damage measured as lactate dehydrogenase (LDH) release [units/L]	Jaqpot 4: http://jaqpot.org/m_detail?name=cm49KGhUjkMw6wyntKQF Jaqpot 5: https://app.jaqpot.org/model/fvAe4KnIOOiNgGf7Ve1p
3	Regression model to understand the aggregated ZVCN against E.Coli by MLR (Placket-Burman design) **Rispoli et al. (2010)**	In vitro - Cytotoxicity - measured as percentage of dead E. Coli population	Jaqpot 4: http://jaqpot.org/m_detail?name=48JYATz0KFTkZjGd8AfS Jaqpot 5: https://app.jaqpot.org/model/8su6n4cfcJpzZD2NDZGN
4	Prediction of the Biological surface adsorption index (BSAI) on different NPs by MLR **Xia et al. (2011)**	log(k) k: adsorption coefficient	Jaqpot 4: http://jaqpot.org/m_detail?name=DjRQk8AqG42nckg5KoxZ Jaqpot 5: https://app.jaqpot.org/model/gSvjUZ17EEAV5OWL7UIs
5	Predictive model of TiO2 NPs damage on membrane cell by SMILES-based optimal descriptor and Monte Carlo technique (CORAL software) **Toropova & Toropov (2013)**	In vitro - Cytotoxicity - membrane damage measured as lactate dehydrogenase (LDH) release [units/L]	Jaqpot 4: http://jaqpot.org/m_detail?name=4oxlwXBZMJ4suYFTSl4d Jaqpot 5: https://app.jaqpot.org/model/nTJgb4Ss3zHIYZEcbg78
6	Cytotoxicity of metal oxide to bacteria E.Coli models by Periodic table-based descriptors and stepwise-MLR **Kar et al. (2014)**	In vitro - Cytotoxicity - measured as pEC50	Jaqpot 4: http://jaqpot.org/m_detail?name=EFffilLYKMgLUq3qNYBw Jaqpot 5: https://app.jaqpot.org/model/QgRRwyU8r7e0NubEuDdX
7	Photo-induced toxicity of metal oxide NPs to E. Coli by MLR (dark condition case) **Pathakoti et al. (2014)**	In vitro - Cytotoxicity - measured as -log(LC50)	Jaqpot 4: http://jaqpot.org/m_detail?name=KIWUeelVM8x7x1iC7cXi Jaqpot 5: https://app.jaqpot.org/model/hygpzrH71XS1Wr8lGS69
8	Photo-induced toxicity of metal oxide NPs to E. Coli by MLR (Photo-induced (light) case) **Pathakoti et al. (2014)**	In vitro - Cytotoxicity - measured as -log(LC50)	Jaqpot 4: http://jaqpot.org/m_detail?name=o6Jr81BfQtUddgmwqaee Jaqpot 5: https://app.jaqpot.org/model/5gCY316DzDh1Fdw4aigo
9	Predicting metal oxide Nps toxicity to E. Coli cell line by MLR **Toropov et al. (2018)**	In vitro - Cytotoxicity - measured as log(1/EC50)	Jaqpot 4: http://jaqpot.org/m_detail?name=qul6HlLHSypXWX8zvMQ3 Jaqpot 5: https://app.jaqpot.org/model/OAiBYuee5PLJ7F580f2J
10	Predicting C60 solubility in organic solvents by SMILES-based optimal descriptor and Monte Carlo technique **Gharagheizi & Alamdari (2008)**	Solubility in organic solvents	Jaqpot 4: http://jaqpot.org/m_detail?name=sCoqY3D3xCpSuyS6RdoQ Jaqpot 5: https://app.jaqpot.org/model/VRp8f6A4DuJc8fsavvpB

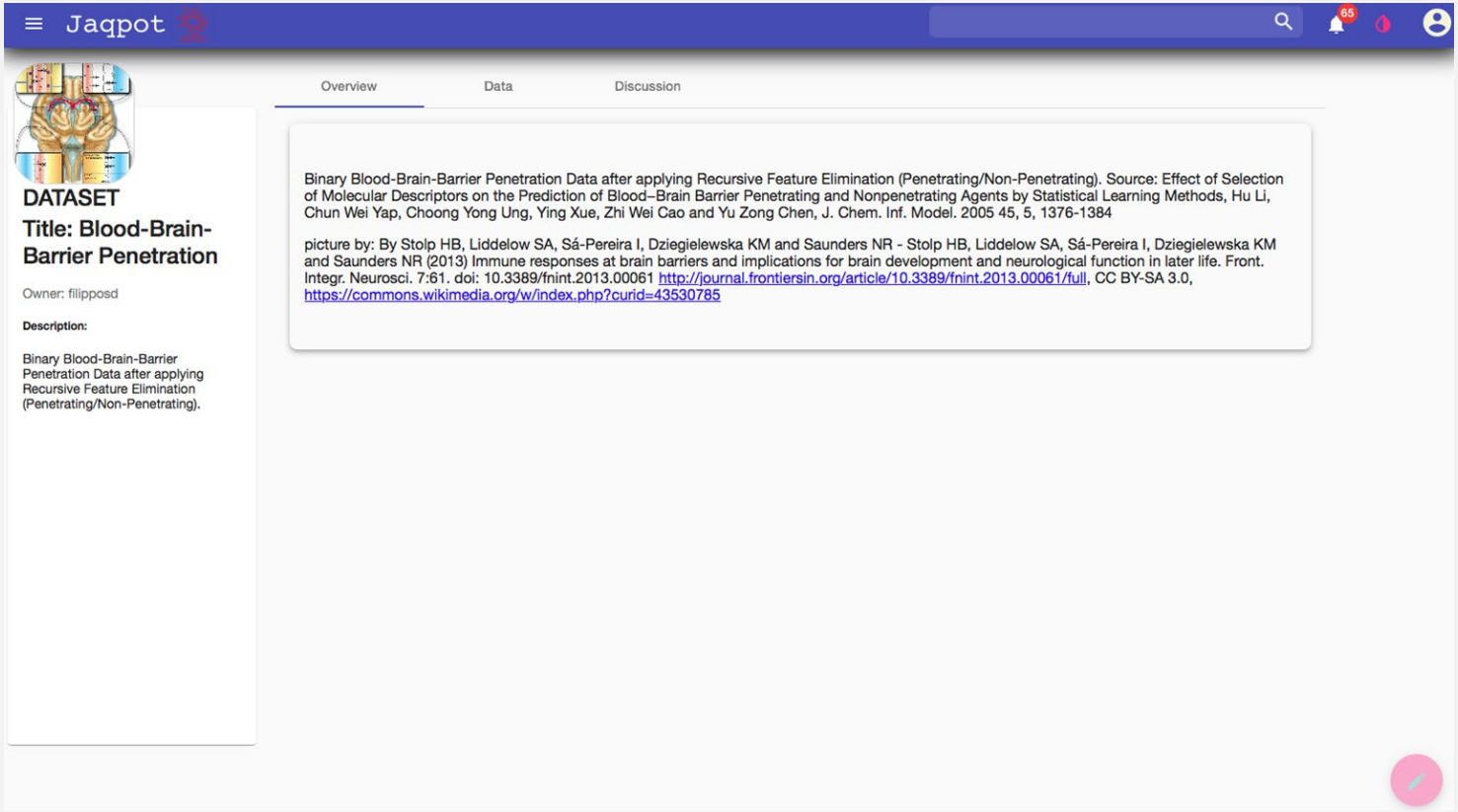
Jaqpot 5 - Datasets - Uploading new dataset



Jaqpot 5 - New dataset -Dataset details

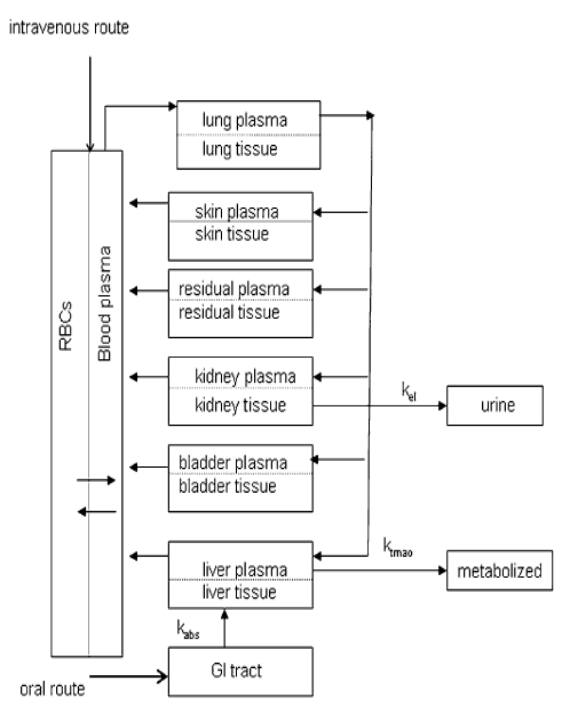


Jaqpot 5 - Dataset page



PBPK model: Multi-compartment model (System of Ordinary Differential Equations, ODEs)

- Mechanistic models that incorporate species' physiology
- Provide information about the concentration-time profile
- Advantage: can conduct meaningful extrapolation
- Two types of parameters: physiological and drug-related



Non-metabolizing compartments:

$$V_{bl}^i \frac{dC_v^i(t)}{dt} = Q_i(C_{art}(t) - C_v^i(t)) - \pi_i\left(C_v^i(t) - \frac{C^i(t)}{P_i}\right)$$

$$V^i \frac{dC^i(t)}{dt} = \pi_i\left(C_v^i(t) - \frac{C^i(t)}{P_i}\right)$$

Metabolizing compartments:

$$V_{bl}^i \frac{dC_v^i(t)}{dt} = Q_i(C_{art}(t) - C_v^i(t)) - \pi_i\left(C_v^i(t) - \frac{C^i(t)}{P_i}\right) - r_{ex}^i\left(C_v^i(t)\right)V_{bl}^i$$

$$V^i \frac{dC^i(t)}{dt} = \pi_i\left(C_v^i(t) - \frac{C^i(t)}{P_i}\right) - r_{met}^i\left(C^i(t)\right)V^i$$

Blood:

$$V^{pl} \frac{dC^{pl}(t)}{dt} = u(t) + \sum_{i \in I_0 \cup I_1} Q_i C_v^i(t) + \pi_{rbc} C^{rbc}(t) - \pi_{pl} C^{pl}(t) - Q_{pl} C^{pl}(t)$$

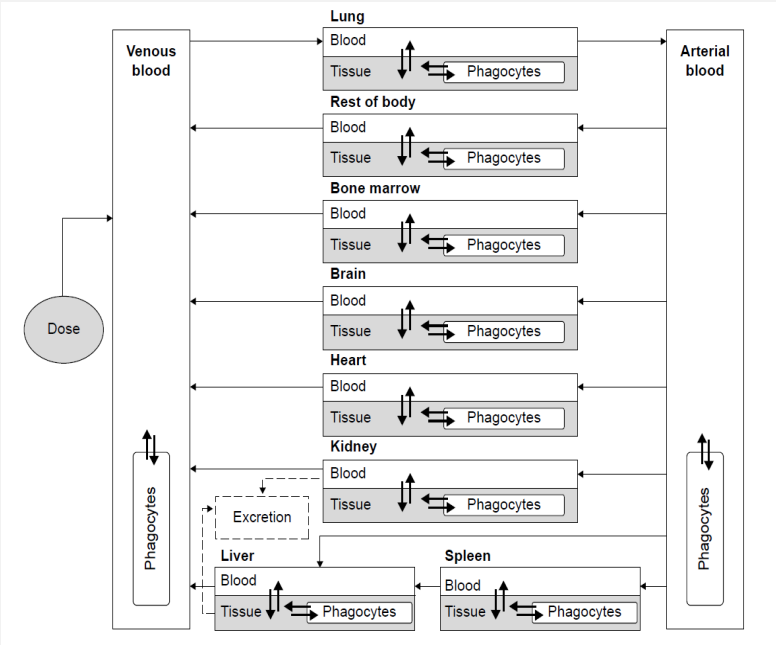
$$V^{rbc} \frac{dC^{rbc}(t)}{dt} = \pi_{pl} C^{pl}(t) - \pi_{rbc} C^{rbc}(t)$$

Lung:

$$V_{bl}^{lu} \frac{dC^{art}(t)}{dt} = Q_{lu}(C^{pl}(t) - C^{art}(t)) - \pi_{lu}\left(C^{art}(t) - \frac{C^{lu}(t)}{P_{lu}}\right)$$

$$V^{lu} \frac{dC^{lu}(t)}{dt} = \pi_{lu}\left(C^{art}(t) - \frac{C^{lu}(t)}{P_{lu}}\right)$$

Physiologically based pharmacokinetics (PBPK) models for nanomaterials



Carlanderg U., Li D., Jolliet O.,, Emond C., Johanson G., Toward a general physiologically-based pharmacokinetic model for intravenously injected nanoparticles, International Journal of Nanomedicine, 2016

The Jaqpotr library: Uploading a model to Jaqpot through R

- Define a dataset containing the dose, infusion time, initial concentrations and other covariates used as model input
- Define a system of differential equations as a function (according to the deSolve package)
- Define a function that generates covariates, whose output will be parsed to the differential equations
- Define the compartment names and declare the input compartment

```
#############
# User input #
#############

weight <- 70 #in kg
gender <- 0 #0 for male | 1 for female
dose <- 10  # in mg
infusion_time <- 5/60 #infusion time in hours
C0_MU <- 0
C0_AD <- 0
C0_GO <- 0
C0_SK <- 0
C0_HT <- 0
C0_BR <- 0
C0_KI <- 0
C0_RE <- 0
C0_ST <- 0
C0_IN <- 0
C0_LU <- 0
C0_LI <- 0
C0_ART <- 0
C0_VEN <- 0
user_input <-data.frame(weight ,gender, dose, infusion_time, C0_MU, C0_AD, C0_GO, C0_SK, C0_HT, C0_BR, C0_KI,
                        C0_RE, C0_ST, C0_IN, C0_LU, C0_LI, C0_ART, C0_VEN )
###################
# compartment names #
###################

comp_names <- c("MU", "AD", "GO" , "SK", "HT", "BR", "KI", "RE", "ST", "IN","LU","LI","ART","VEN")
```

```
#############################################################
# Covariate model for calculation of organ flows and volumes #
#############################################################

covariates  <- function(w,gender){

  density <- rep(1,14)
  flow_frac<-c(0.17,0.05,0.001,0.05,0.04,0.12,0.19,0,0.01,0.14,1,0.065)

  const<-list(k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11,k12,k13,k14)
  const_w<-list(k1_w,k2_w,k3_w,k4_w,k5_w,k6_w,k7_w,k8_w,k9_w,k10_w,k11_w,k12_w,k13_w,k14_w
  vol<-rep(0,14)
  flow<-rep(0,12)
  TF<-(187*(w)^0.81)*60/1000 #Total flow
  weight <- w

  if (gender==0){
    if (w>75){
      w<-75*1000
    }
    else{
      w<-w*1000
    }
    co<-const
    for (i in 1:14){

      vol[i]<-(co[[i]][1]+co[[i]][2]*w+co[[i]][3]*w^2+co[[i]][4]*w^3+co[[i]][5]*w^4+co[[1
      vol[i]<-vol[i]*weight/density[i]
    }
  }

  else if (gender==1){
    if (w>65){
      w<-65*1000
    }
    else{
      w<-w*1000
    }
    co<-const_w
    for (i in 1:14){

      vol[i]<-(co[[i]][1]+co[[i]][2]*w+co[[i]][3]*w^2+co[[i]][4]*w^3+co[[i]][5]*w^4+co[[1
      vol[i]<-vol[i]*weight/density[i]
    }
  }
  vol[8]<-weight-sum(vol)
  combine<-c(flow,vol)

  return(combine)
}
```

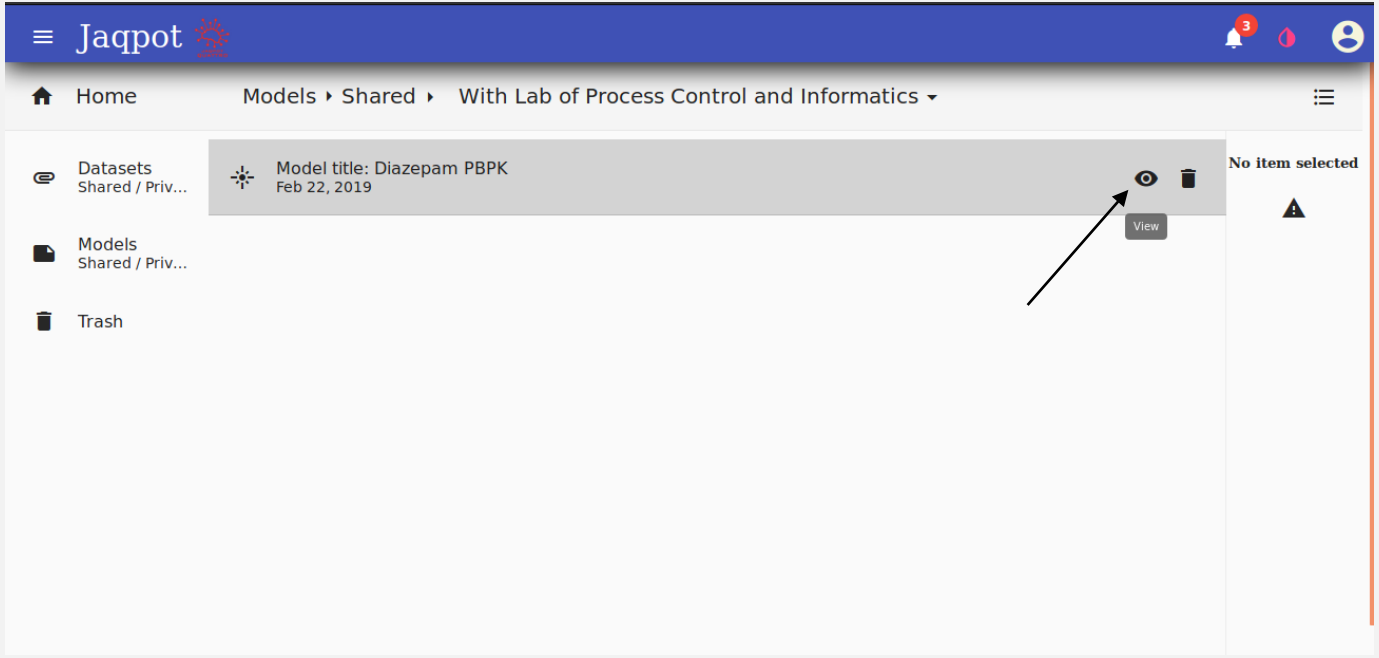
```
################
# ODEs function #
################

odes<-function(time,C,params){

  dCdt<-rep(0,14)

  Q_MU<-params[1]
  Q_AD<-params[2]
  Q_TE<-params[3]
  Q_SK<-params[4]
  Q_HT<-params[5]
  Q_BR<-params[6]
  Q_KI<-params[7]
  Q_RE<-params[8]
  Q_ST<-params[9]
  Q_SPL<-params[10]
  Q_LU<-params[11]
  Q_LI<-params[12]
  Q_ART<-params[11]
  Q_VEN<-params[11]



  dCdt[1]<-(-Q_MU*C[1]/(KP_MU*V_MU))+(Q_MU*C[13]/V_MU)
  dCdt[2]<-(-Q_AD*C[2]/(KP_AD*V_AD))+(Q_AD*C[13]/V_AD)
  dCdt[3]<-(-Q_TE*C[3]/(KP_TE*V_TE))+(Q_TE*C[13]/V_TE)
  dCdt[4]<-(-Q_SK*C[4]/(KP_SK*V_SK))+(Q_SK*C[13]/V_SK)
  dCdt[5]<-(-Q_HT*C[5]/(KP_HT*V_HT))+(Q_HT*C[13]/V_HT)
  dCdt[6]<-(-Q_BR*C[6]/(KP_BR*V_BR))+(Q_BR*C[13]/V_BR)
  dCdt[7]<-(-Q_KI*C[7]/(KP_KI*V_KI))+(Q_KI*C[13]/V_KI)
  dCdt[8]<-(-Q_RE*C[8]/(KP_RE*V_RE))+(Q_RE*C[13]/V_RE)
  dCdt[9]<-(-Q_ST*C[9]/(KP_ST*V_ST))+(Q_ST*C[13]/V_ST)
  dCdt[10]<-(-Q_SPL*C[10]/(KP_SPL*V_SPL))+(Q_SPL*C[13]/V_SPL)
  dCdt[11]<-(-Q_LU*C[11]/(KP_LU*V_LU))+(Q_VEN*C[14]/V_LU)
  dCdt[12]<-(Q_HA*C[13]/V_LI)+(Q_ST*C[9]/(KP_ST*V_LI))+(Q_SPL*C[10]/(KP_SPL*V_LI))-(((Q_H+CL*f_UB)/(KP_LI*V_LI))*C[12])
  dCdt[13]<-(-Q_ART*C[13]/V_ART)+(Q_LU*C[11]/(V_ART*KP_LU))

  if (time<=T_inf){
    dCdt[14]<-(-Q_LU*C[14]/V_VEN)+(Q_H*C[12]/(KP_LI*V_VEN))+(Q_KI*C[7]/(KP_KI*V_VEN))+
      (Q_MU*C[1]/(KP_MU*V_VEN))+(Q_AD*C[2]/(KP_AD*V_VEN))+(Q_SK*C[4]/(KP_SK*V_VEN))+
      (Q_TE*C[3]/(KP_TE*V_VEN))+(Q_HT*C[5]/(KP_HT*V_VEN))+(Q_BR*C[6]/(KP_BR*V_VEN))+
      (Q_RE*C[8]/(KP_RE*V_VEN))+(dose/(V_VEN*1000*T_inf))  #units are in ng/mL
  }
  else{

    dCdt[14]<-(-Q_LU*C[14]/V_VEN)+(Q_H*C[12]/(KP_LI*V_VEN))+(Q_KI*C[7]/(KP_KI*V_VEN))+
      (Q_MU*C[1]/(KP_MU*V_VEN))+(Q_AD*C[2]/(KP_AD*V_VEN))+(Q_SK*C[4]/(KP_SK*V_VEN))+
      (Q_TE*C[3]/(KP_TE*V_VEN))+(Q_HT*C[5]/(KP_HT*V_VEN))+(Q_BR*C[6]/(KP_BR*V_VEN))+
      (Q_RE*C[8]/(KP_RE*V_VEN))
  }

  list(dCdt)
```

Calling deploy.pbpk()

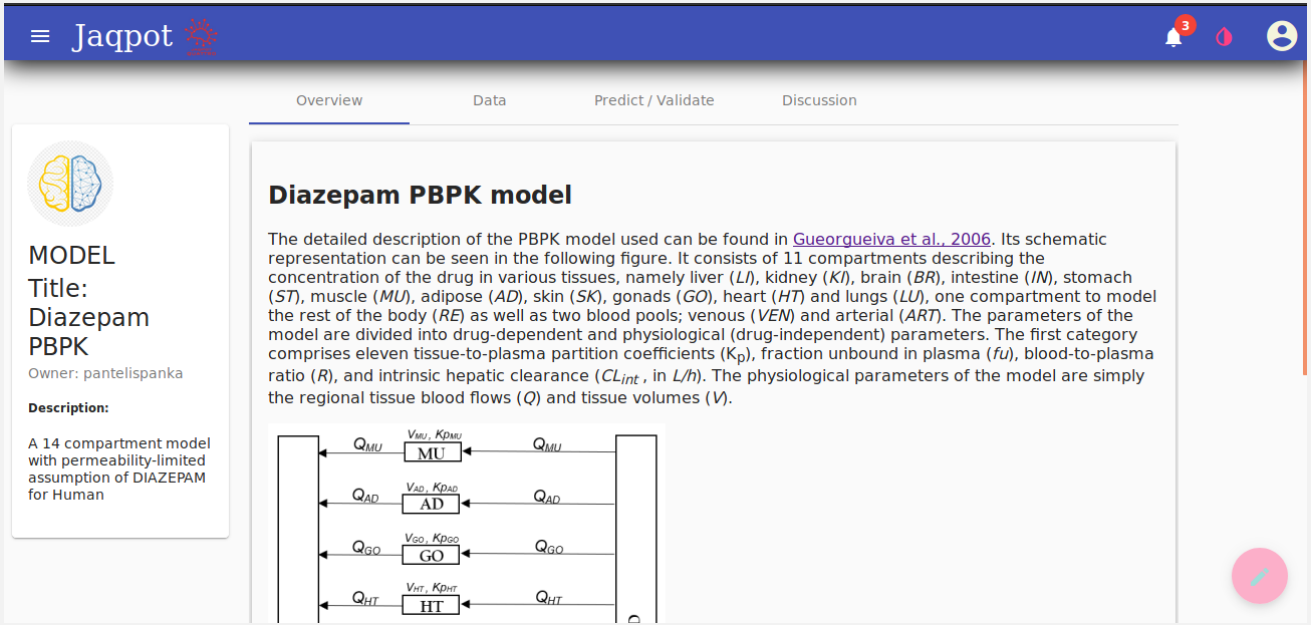
- Load jaqpotr
- Call deploy.pbpk() and provide the dataset, covariate model, differential equations, compartment names and input
- Give Jaqpot base path (https://api.jaqpot.org/) and choose the authentication method (Api key or Jaqpot credentials, i.e. username and password)
- Add a model title (cannot be edited) and a short description (editable)
- The PBPK model is generated and the model id is provided.
- The user can further refine the model description and information through the Jaqpot UI

```
> deploy.pbpk(data=user_input, odes=odes, comp.names=comp_names, comp.in='VEN', cov.model=covariates )
Base path of jaqpot *e.g.: https://api.jaqpot.org/ : https://api.jaqpot.org/
Please choose authentication method ([1]=login / [2]=Provide Api Key): 1
Username: PeriklisTs
Title of the model: upload test
Short description of the model:this is a short description
[1] "Model created. The id is:  DLGxOzkvGkU9ctHi4wRW . Please visit the application to further document your
 model."
>
```

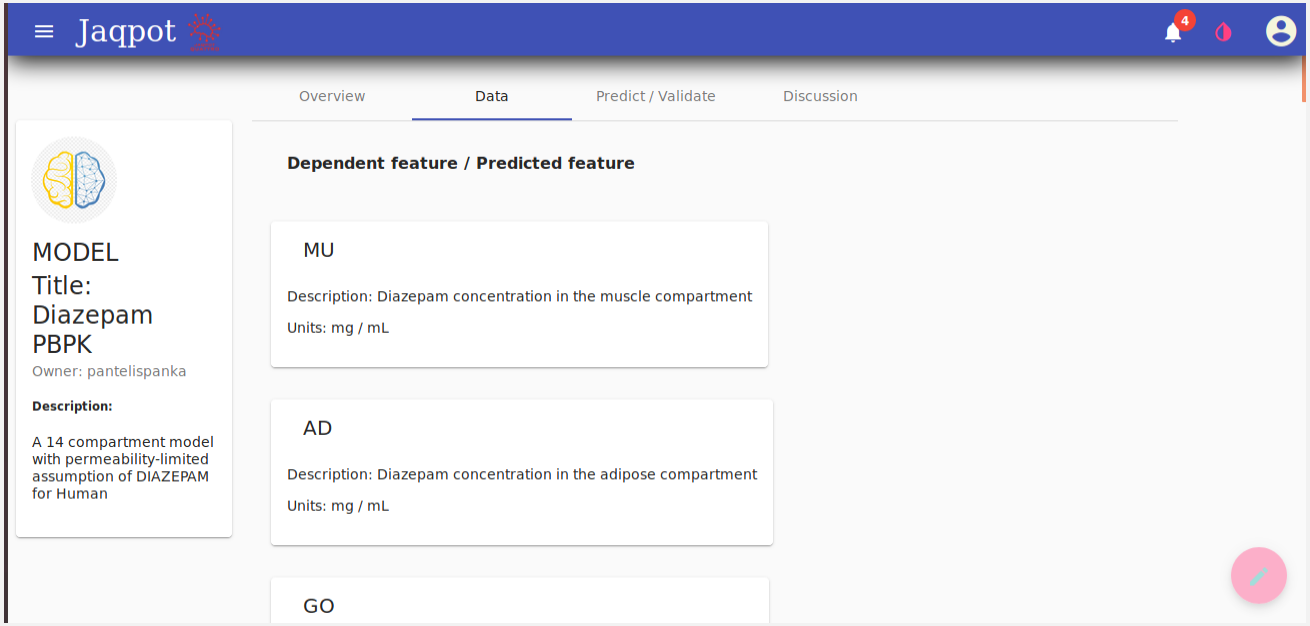

Accessing the model



Entering model information through Jaqpot UI

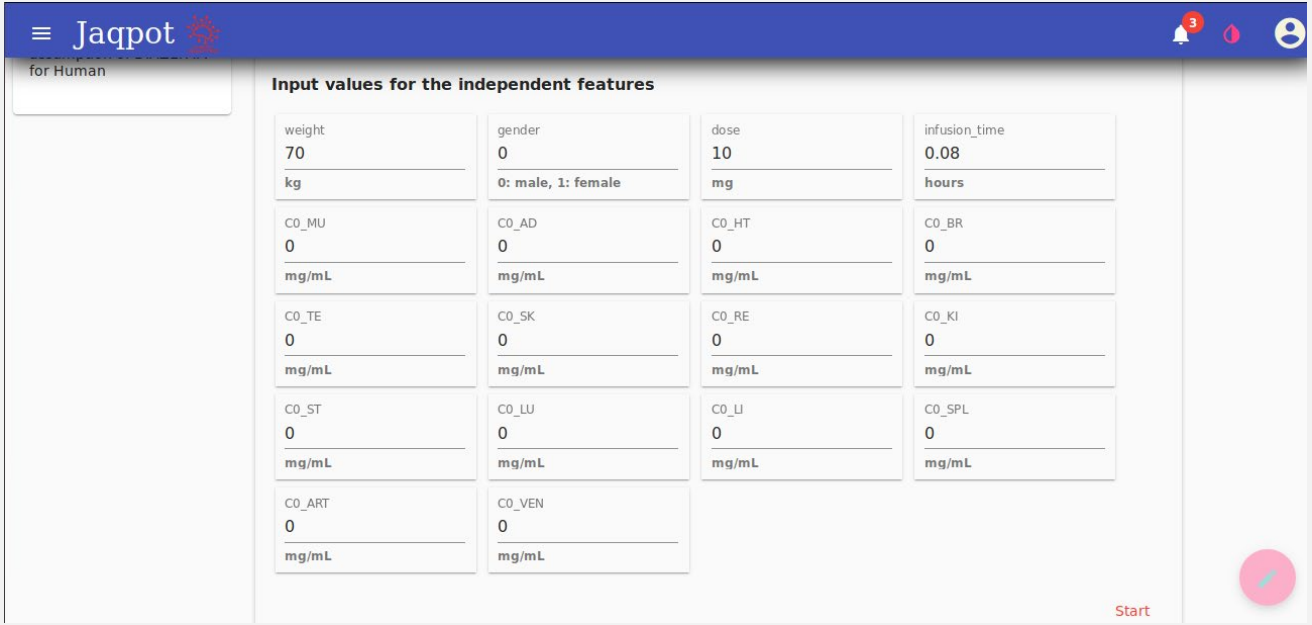


Add description and units to the model variables



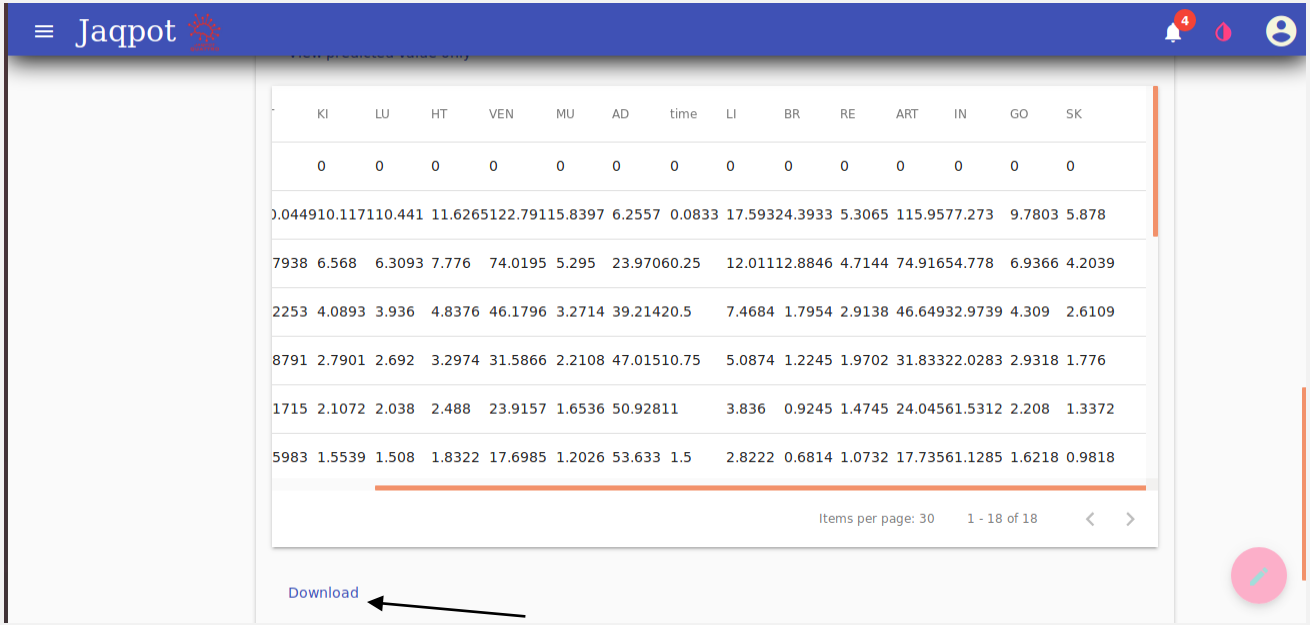
Perform PBPK simulations

- Users can enter dose and physiological characteristics and get predictions
- Upload dataset or manually fill in values

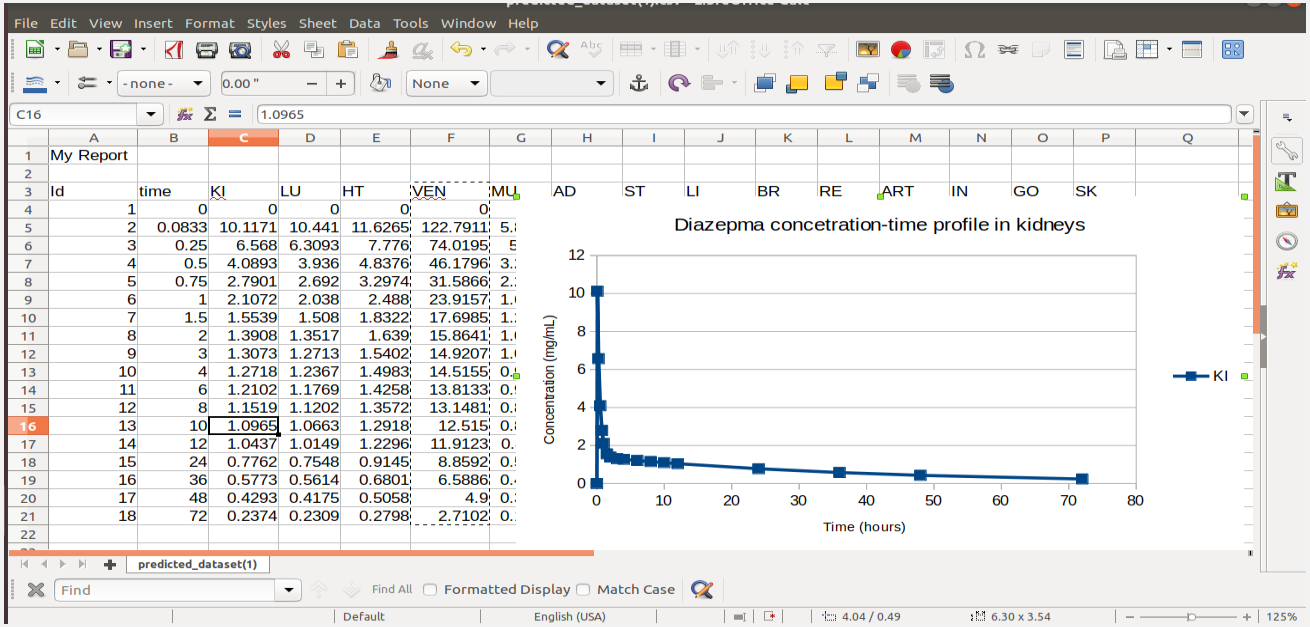


Getting predictions

- Predictions are generated in a table form and can be downloaded as a csv file and be further processed offline



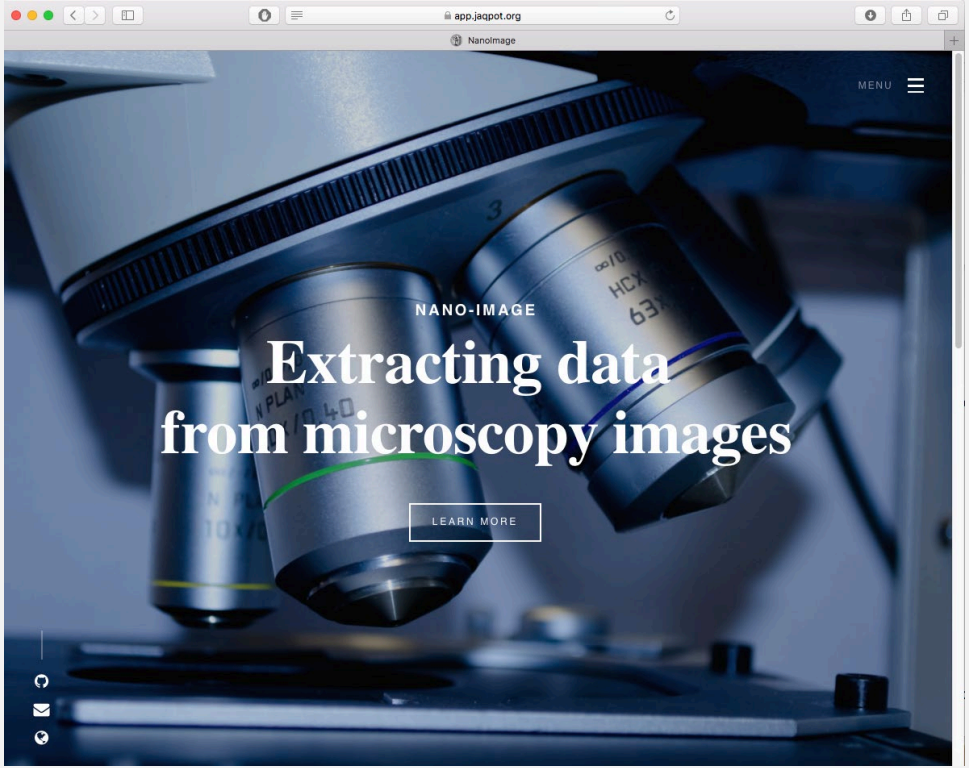
Predictions in a csv file



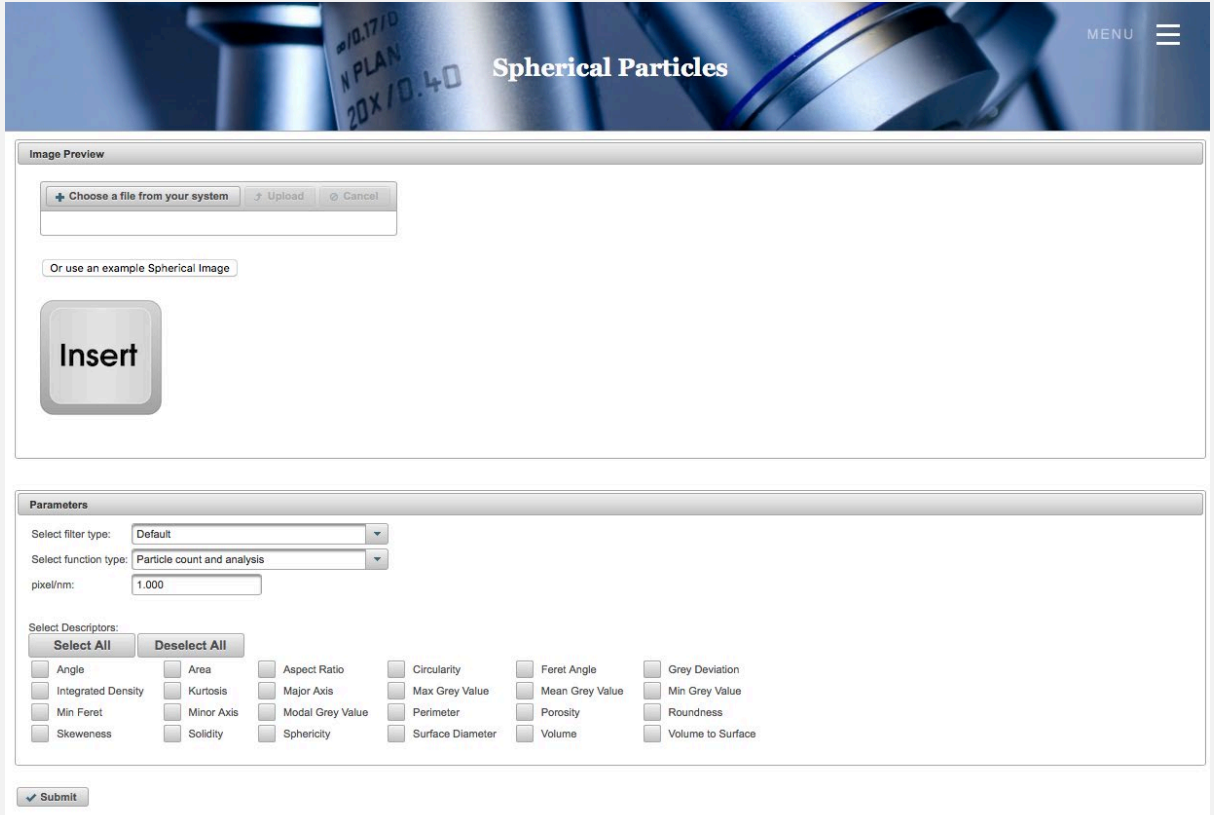
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	My Report																
2																	
3	Id	time	KI	LU	HT	VEN	MU	AD	ST	LI	BR	RE	ART	IN	GO	SK	
4	1	0	0	0	0	0											
5	2	0.0833	10.1171	10.441	11.6265	122.7911	5.										
6	3	0.25	6.568	6.3093	7.776	74.0195	5										
7	4	0.5	4.0893	3.936	4.8376	46.1796	3.										
8	5	0.75	2.7901	2.692	3.2974	31.5866	2.										
9	6	1	2.1072	2.038	2.488	23.9157	1.										
10	7	1.5	1.5539	1.508	1.8322	17.6985	1.										
11	8	2	1.3908	1.3517	1.639	15.8641	1.										
12	9	3	1.3073	1.2713	1.5402	14.9207	1.										
13	10	4	1.2718	1.2367	1.4983	14.5155	0.										
14	11	6	1.2102	1.1769	1.4258	13.8133	0.										
15	12	8	1.1519	1.1202	1.3572	13.1481	0.										
16	13	10	1.0965	1.0663	1.2918	12.515	0.										
17	14	12	1.0437	1.0149	1.2296	11.9123	0.										
18	15	24	0.7762	0.7548	0.9145	8.8592	0.										
19	16	36	0.5773	0.5614	0.6801	6.5886	0.										
20	17	48	0.4293	0.4175	0.5058	4.9	0.										
21	18	72	0.2374	0.2309	0.2798	2.7102	0.										
22																	

Jaqpot Image Analysis

https://app.jaqpot.org/nanoImage/

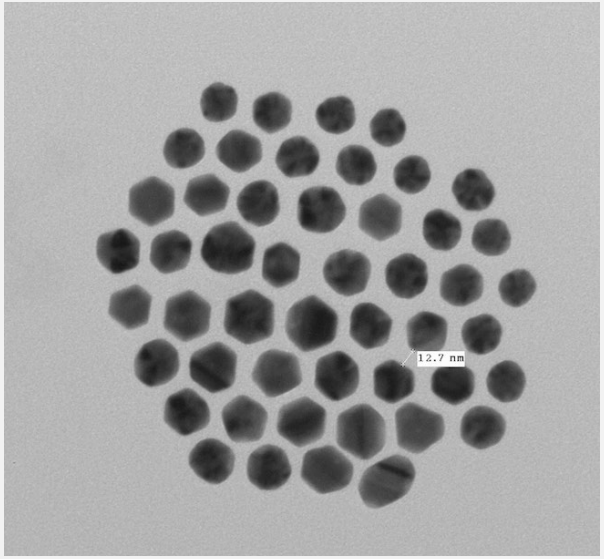


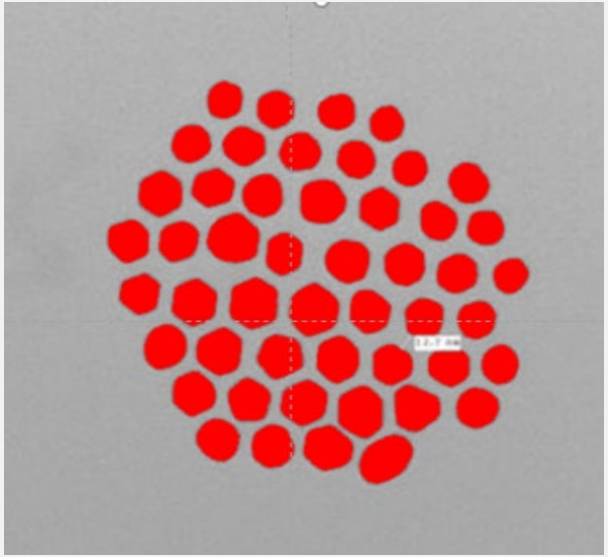
Calculating image descriptors for spherical nanomaterials

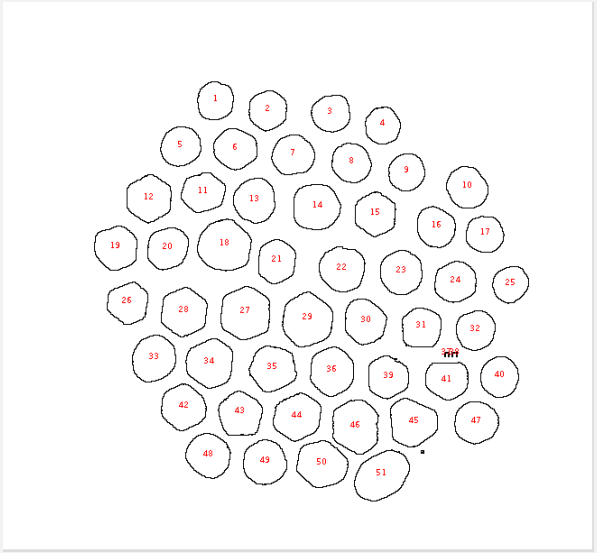


Results of this tool are derived from ImageJ, with additional descriptors provided, as described in:
Philip Doganis, Georgia Tsiliki, Haralambos Sarimveis, Haralambos Chomenidis, Georgios Drakakis, Egon Willighagen, Barry Hardy. (2015, April 29). Deliverable Report D4.2 Descriptor Calculation Algorithms and Methods. Zenodo. http://doi.org/10.5281/zenodo.375609

Calculating image descriptors for spherical nanomaterials







Calculating image descriptors for spherical nanomaterials

List of Properties

Columns

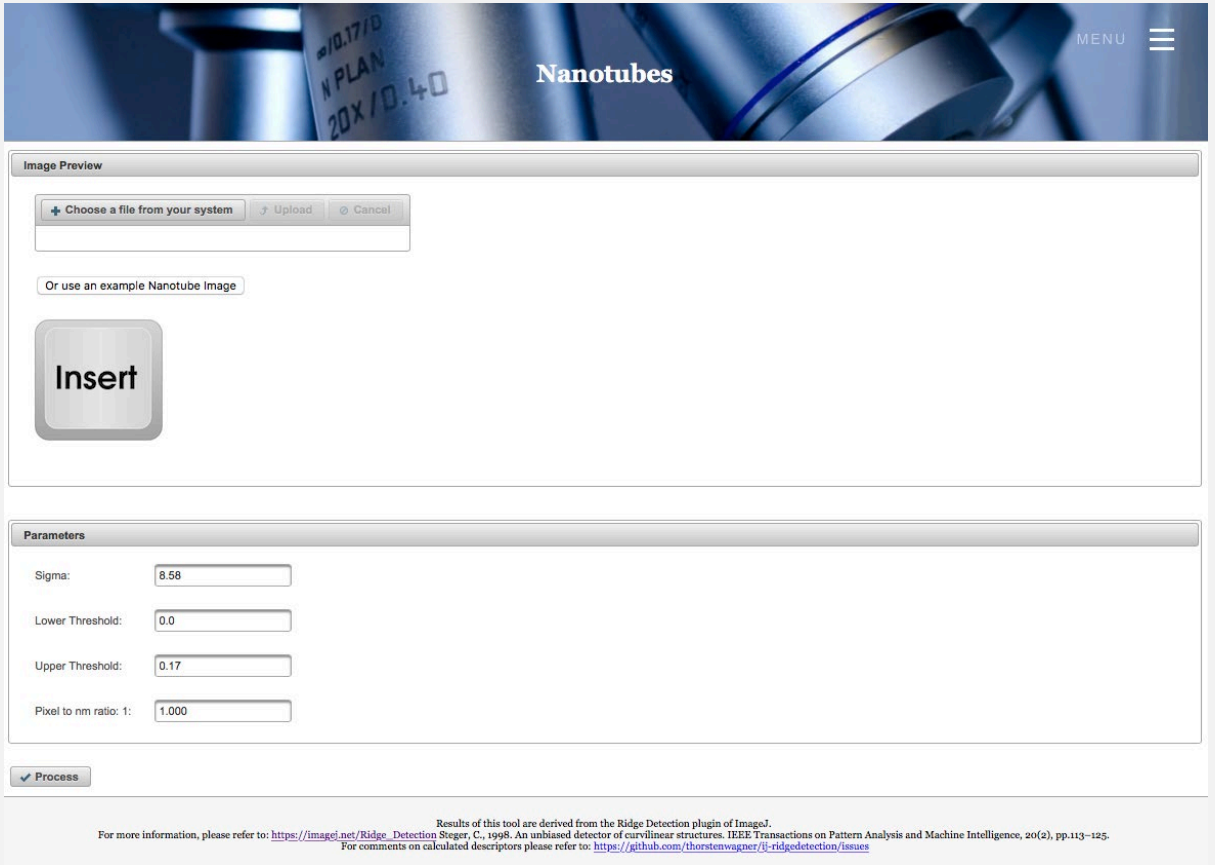
(1 of 3) 1 2 3 25

id	Angle	Area	Aspect Ratio	Circularity	Feret Angle	Grey deviation	Integrated Density	Kurtosis	Major Axis	
Average Particle	96.13313	780.2823	1.0759023	0.8592815	97.67603	21.327768	53979.883	-0.44196674	32.50095	1:
1	31.853	514.877	1.049	0.889	22.620	20.576	39233.001	-0.810	26.222	1:
2	23.957	552.505	1.061	0.846	158.587	16.775	46490.280	-0.733	27.320	1:
3	4.397	476.167	1.058	0.862	170.910	24.161	33538.219	-0.988	25.329	1:
4	110.551	433.125	1.013	0.886	111.801	20.821	35176.785	-0.698	23.633	1:
5	106.344	813.192	1.125	0.852	103.627	24.343	50797.166	-0.422	34.125	1:
6	131.483	631.550	1.099	0.904	109.654	23.437	43264.853	-0.708	29.726	1:
7	84.296	559.543	1.074	0.841	82.875	24.121	40191.019	-1.008	27.661	1:
8	25.308	655.643	1.089	0.887	32.106	22.161	44481.122	-0.506	30.148	1:
9	106.995	769.609	1.190	0.857	94.205	19.459	58704.946	-0.626	34.141	1:
10	138.673	523.269	1.112	0.880	162.582	19.736	39727.846	-0.525	27.217	1:
11	170.300	851.091	1.067	0.868	147.724	26.882	44428.605	-0.206	34.011	1:
12	162.309	707.347	1.032	0.845	142.943	20.357	50640.700	-0.664	30.490	1:
13	124.555	693.000	1.049	0.862	146.310	20.356	49973.146	-0.491	30.430	1:
14	70.426	836.202	1.154	0.872	52.633	22.376	51461.472	-0.118	35.048	1:
15	143.829	845.406	1.073	0.884	110.556	22.381	54884.242	-0.403	33.977	1:
16	63.841	531.119	1.031	0.889	23.629	20.101	37434.179	-0.321	26.400	1:
17	46.369	602.044	1.141	0.857	23.070	20.942	47170.286	-0.910	29.568	1:
18	51.225	743.621	1.028	0.861	79.046	22.626	52225.667	-0.724	31.202	1:
19	33.415	624.241	1.042	0.881	40.061	22.603	40840.706	-0.457	28.774	1:
20	58.898	705.994	1.056	0.885	51.633	24.296	46138.366	-0.647	30.816	1:
21	57.115	770.421	1.071	0.894	52.595	21.787	50777.405	-0.292	32.416	1:
22	79.090	870.852	1.100	0.824	81.870	14.121	76792.515	-0.466	34.924	1:
23	140.855	832.412	1.018	0.836	80.538	21.035	62232.207	-0.601	32.850	1:
24	104.264	1026.235	1.136	0.863	84.668	23.455	59227.132	0.062	38.526	1:

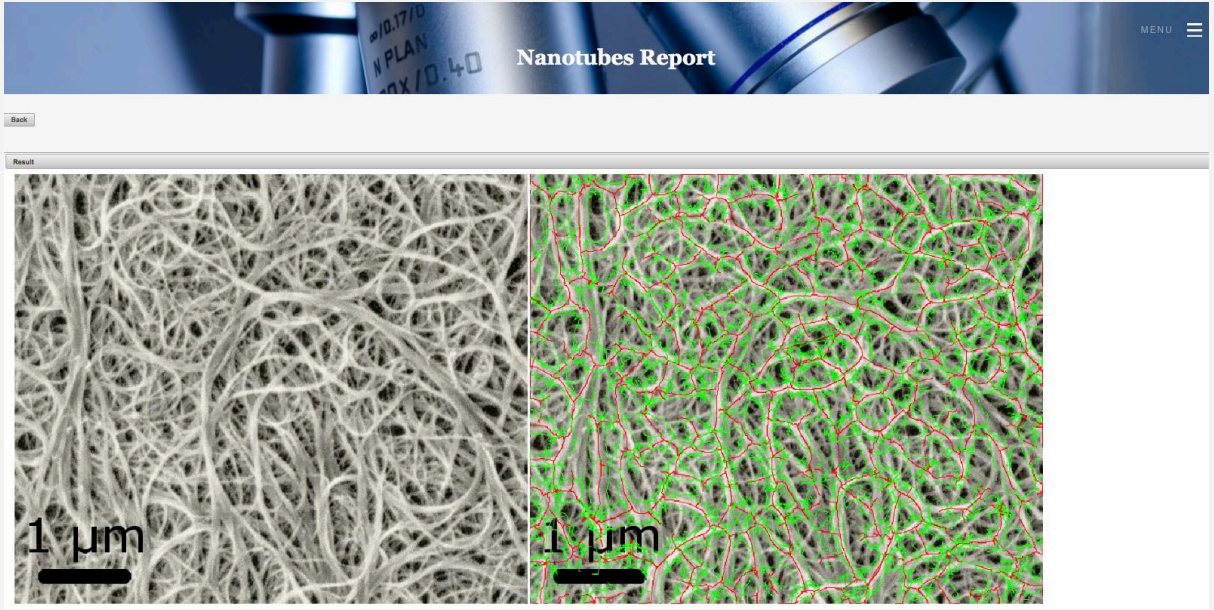
(1 of 3) 1 2 3 25

Download Spherical Descriptors: XLS PDF CSV XML

Calculating descriptors for carbon nanotubes



Calculating descriptors for carbon nanotubes



Calculating descriptors for carbon nanotubes

MENU

Columns

Summary per Nanotube

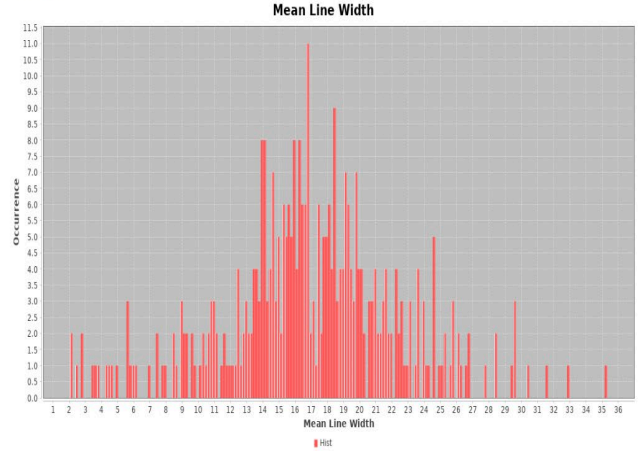
(1 of 18) 1 2 3 4 5 6 7 8 9 10 25

Contour Id.	Length	Mean Line Width
Average	48.454	19.366
1898	19.418	28.483
1879	17.790	24.792
2763	7.478	25.183
2149	46.242	25.178
2294	14.275	16.801
3027	32.286	12.935
1903	14.343	26.992
2919	13.759	1.604
2971	83.469	20.631
2962	90.604	16.003
2227	7.791	10.702
2958	21.103	21.867
2774	9.903	9.732
2986	81.181	24.179
2207	16.636	26.074
2977	10.734	10.881
2781	17.560	30.725
2235	4.866	26.727
2160	13.263	28.741
2243	30.053	18.465
1654	30.000	26.611
2916	89.276	11.774
1867	40.501	23.434
1897	30.406	27.627

(1 of 18) 1 2 3 4 5 6 7 8 9 10 25

Download Summ

Mean Width Histogram



Jaqpot: Other services

ToxFlow[1]**:** a read-across methodology, considers the multi-perspective characterization of nanoparticles and selects neighbour based on both physicochemical and biological similarity criteria, http://147.102.86.129:3838/toxflow[1]

Dose-response modelling : a web implementation of the Benchmark Dose (BMD) approach for estimating derived no-effect levels (DNELs). Integrates the PROAST open-source software developed by RIVM

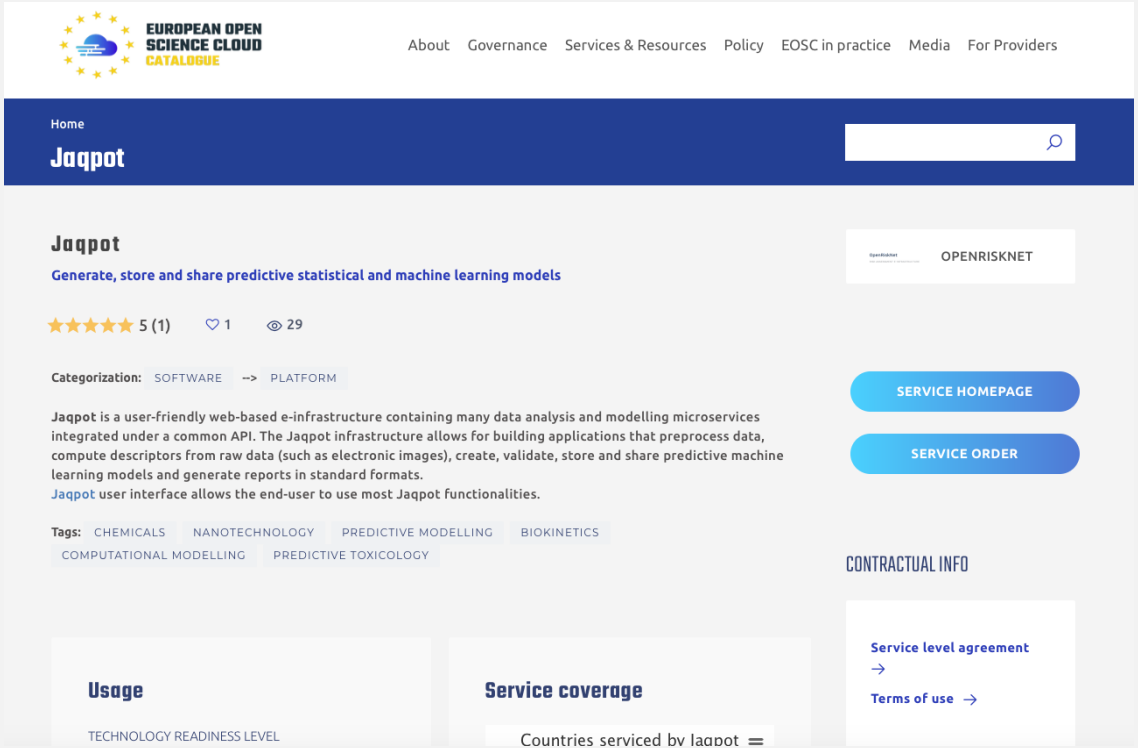
Biodescriptor calculations[2] **:** descriptors which are derived by enriching high-throughput omics data with biological-pathway information

Biokinetics: Integration of high-throughput toxicokinetics (httk) and PK-Sim software platforms

[1]Varsou DD, Tsiliki G, Nymark P, Kohonen P, Grafström R, Sarimveis H.(2018) toxFlow: A Web-Based Application for Read-Across Toxicity Prediction Using Omics and Physicochemical Data, J Chem Inf Model. 2018 Mar 26;58(3):543-549. doi: 10.1021/acs.jcim.7b00160

[2]Tsiliki G, Nymark P, Kohonen P, Grafström R, Sarimveis H. (2017) Enriching Nanomaterials Omics Data: An Integration Technique to Generate Biological Descriptors, Small Methods; 1700139.

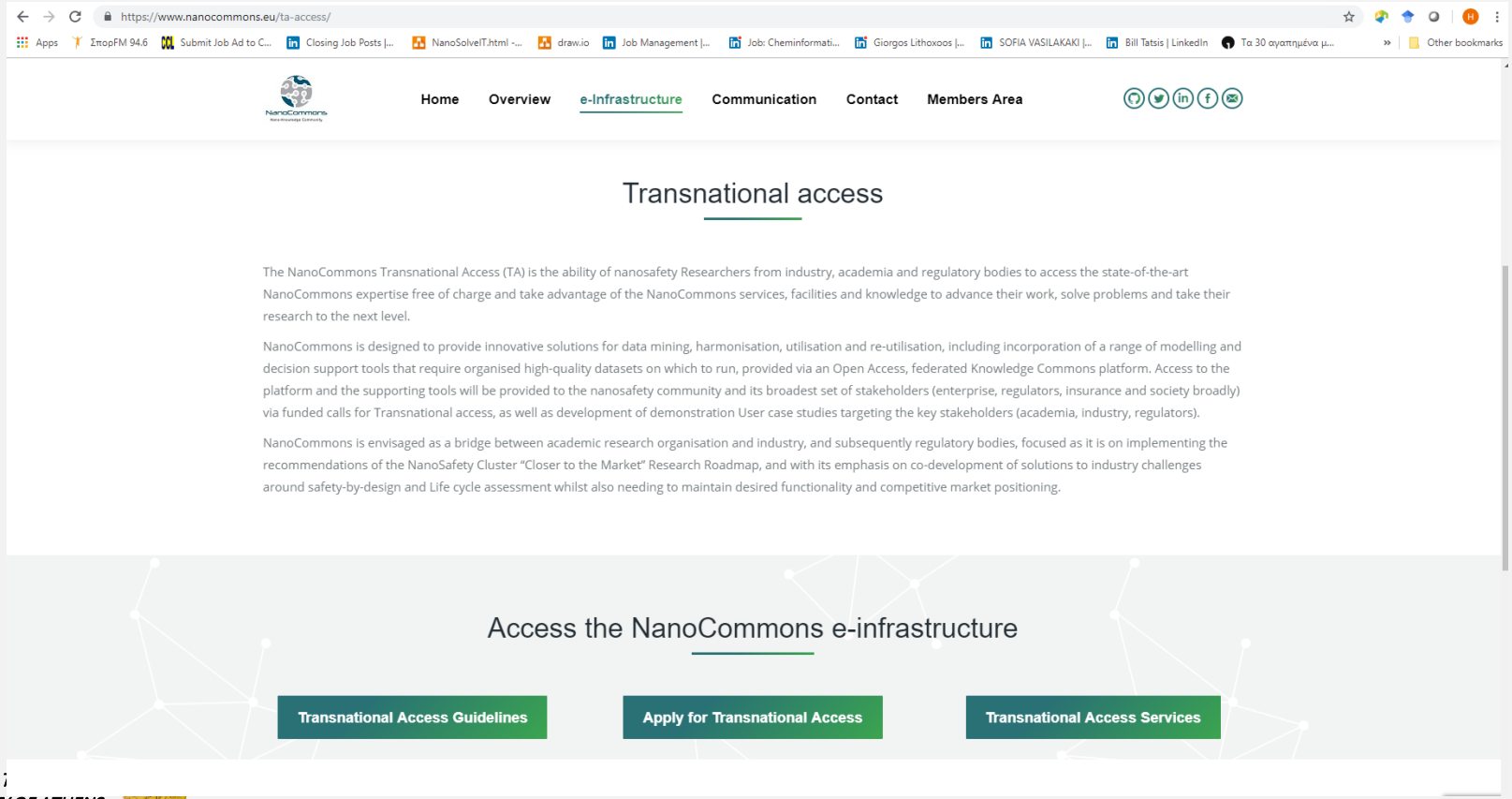
Jaqpot in EOSC Catalogue



https://catalogue.eosc-portal.eu/service/openrisknet.jaqpot_

NanoCommons Transnational Access Programme

https://www.nanocommons.eu/apply-for-access/



Conclusions

Jaqpot constitutes a **universal platform**
that **produces and exchanges** semantically annotated datasets
and machine learning **predictive models** .

- Jaqpot is an **open source web infrastructure** with a flexible structure:
  - **generalization across disciplines**
  - **integration to third-party tools** in order to fulfil modelling needs on the Cloud
  - **easy integration** to diverse **architectures/ontologies/knowledge domains**.
- Jaqpot offers a seamless way to:
  - **take models out of the desktop or paper** and automatically make them available as web services via their URI
  - functionality is provided both through an **API and GUIs**.

Thank you!