DELIVERABLE REPORT D4.1

Analysis and Modelling Specifications

Deliverable Dx-y

GRANT AGREEMENT:	604134
ACRONYM:	eNanoMapper
NAME:	eNanoMapper - A Database and Ontology Framework for Nanomaterials Design and Safety Assessment
PROJECT COORDINATOR:	Douglas Connect GmbH
START DATE OF PROJECT; DURATION:	1 February 2014; 36 months
PARTNER(s) RESPONSIBLE FOR THIS DELIVERABLE:	NTUA
DATE:	31.1.2015

VERSION:	[V.5.0.]

Call identifier	FP7-NMP-2013-SMALL-7
Document Type	Deliverable Report
WP/Task	The eNanoMapper analysis and modelling infrastructure will be built upon the OpenTox web services framework following the principles of the Representational State Transfer (REST) design model, which is a well-established software architecture for distributed applications. This task performs the necessary adjustments and extension to OpenTox ontology and APIs for descriptor calculation, clustering, predictive model building and model validation. It also specifies the data analysis and modelling algorithms and methods that will be implemented to automate the meta-data analysis of results and support users with obtaining meaningful conclusions in regard to the underlying chemical and biological mechanisms of human and environmental toxicity of ENMs.
Document ID	eNanoMapper D4.1
Status	Final

Partner Organisations	● Douglas Connect, GmbH (DC) ● National Technical University of Athens (NTUA) ● In Silico Toxicology (IST) ● Ideaconsult (IDEA) ● Karolinska Institutet (KI) ● VTT Technical Research Centre of Finland (VTT) ● European Bioinformatics Institute (EMBL-EBI) ● Maastricht University (UM)

Authors	Philip Doganis Georgia Tsiliki Haralambos Sarimveis Vedrin Jeliazkov
Purpose of the Document	To report on the technical specifications that have been set for the update of OpenTox API infrastructure, modelling tools, and descriptor calculations.
Document History	1.Table of Contents, 13/11/2014 2.First draft, 17/11/2014 3. Second draft, 28/12/2014 4. Third draft, 14/01/2015 5. Fourth draft, 29/01/2015 6. Final, 31/01/2015

TABLE OF CONTENTS

TABLE OF FIGURES

GLOSSARY

Abbreviation / acronym	Description
3D	3-Dimensional
ANOVA	Analysis of Variance
API	Application Programming Interface
BP	Biological Processes
CC	Cellular Components
CV	Cross-Validation
DFT	Density Functional Theory
DoA	Domain of Applicability
ENM(s)	Engineered Nanomaterial(s)
FFNN	Feed Forward Neural Network
GNU GPL	GNU General Public License
GO	Gene Ontology
GPW	Gaussian and plane waves
GSEA	Gene Set Enrichment Analysis
HC	Hierarchical Clustering
HF	Hartree-Fock
HOMO	Highest Occupied Molecular Orbital
HTTP	Hypertext Transfer Protocol
KEGG	Kyoto Encyclopedia of Genes and Genomes
kNN	K Nearest Neighbour
KS-DFT	Kohn-Sham density functional theory
LASSO	Least Absolute Shrinkage and Selection Operator
LC-MS/MS	Liquid chromatography–Mass Spectrometry/ Mass Spectrometry
LM	Linear Model
LOO	Leave-One-Out
LUMO	Lowest Unoccupied Molecular Orbital
MF	Molecular Functions
MLR	Multiple Linear Regression
MS	Mass Spectrometry
MSigDB	Molecular Signature Databases
NanoQSAR , nQSAR	Nano Quantitative Structure Activity Relationship
OLS	Ordinary Least Squares
OpenTox	http://www.opentox.org/
PLS	Partial Least Squares
PMML	Predictive Model Markup Language
PubMed	http://www.ncbi.nlm.nih.gov/pubmed
QM	Quantum Mechanics
QSAR	Quantitative Structure Activity Relationship

RBF	Radial Basis Function
REST	Representational State Transfer
RF	Random Forest
RFE	Recursive Feature Elemination
RMSE	Root Mean Square Error
RSS	Rich Site Summary
SVM	Support Vector Machines
TEM	Transmission Electron Microscopy
UniProt	Universal Protein Database
URI	Uniform Resource Identifier
VIP	Very Important Person
WS	Web Services

1. EXECUTIVE SUMMARY

The eNanoMapper project aims to build an ontology and database to collate and describe data relevant for the development of “safe by design” Engineered Nanomaterials (ENMs). Work Package 4 will develop computational infrastructure capable to analyse and extract knowledge out of diverse types of ENM-related theoretical descriptors, experimental data and associated metadata, including provenance of experimental data and experimental conditions and protocols. This deliverable report defines the technical specifications of the eNanoMapper analysis and modelling infrastructure. More specifically, it describes the necessary adjustments to OpenTox APIs for descriptor calculation, clustering, predictive model building and model utilization and specifies the data analysis and modelling algorithms, methods and tools that will be implemented throughout the project to automate the meta-data analysis of results and to support users with obtaining meaningful conclusions in regard to the underlying chemical and biological mechanisms of toxicity of ENMs.

2. INTRODUCTION

In contrast with the well-developed approaches in modelling chemicals properties, modelling ENMs presents distinct challenges and ENM similarity must accommodate many additional aspects (Winkler, et al., 2013; Burello and Worth, 2011; Puzyn et al., 2011; Gajewicz et al., 2012; Fourches et al., 2010). ENMs are not a distinct class of chemical substances, but a rather heterogeneous group, which comprises several classes of core chemistries, sizes and structures (Malkiewicz et al., 2011). Most importantly, there is no canonical representation of the structures and hence the existing molecular descriptors are largely not applicable; however, they could be used to characterise functionalised ENM. Besides quantum effects, surface effects are also observed, due to the surface to volume ratio increasing with decreasing particle size, which affects the reactivity and phase transition temperatures (Roduner, 2006). Nanoparticle coatings can further modify the material properties. Biological interactions of ENMs can also be affected by the exchange between agglomerated and dispersed forms, as well as by the formation of lipid and protein coronas (Monopoli et al., 2011; Kapralov et al., 2012), which may be modified as the ENM moves from one compartment to another.

In order to reflect the required adequate description and supramolecular pattern of ENMs and meet the challenging requirements of modelling ENMs, the OpenTox modelling resources are extended in WP4 of the eNanoMapper project, by integrating and extracting knowledge out of a large number of diverse features, including structural characteristics, spectral information, images, high throughput screening and omics data. The modelling tools should be able to assess potential risks of ENMs, provide information in regard to the underlying chemical and biological toxicity mechanisms, prioritize ENMs for experimental testing, and contribute to the development of ‘safe-by-design’ ENMs, as the potential toxicity and environmental impact of ENMs will be predicted during the design phase. Following the OpenTox architecture, implementation of eNanoMapper modelling software components is based on interoperable, standards-compliant and modular web services maximising cross-talk and interaction between different and diverse sources of data, following the principles of the Representational State Transfer (REST) design model, which is a well-established software architecture for distributed applications.

This deliverable defines the technical specifications that will be used throughout the project to develop the modelling tools so that we will meet the goals of analysing and extracting knowledge out of diverse types of ENM-related theoretical descriptors, experimental data and associated metadata and supporting mechanism-of-action modelling approaches. The report contains four main sections that are briefly described next:

1) OpenTox API extensions: OpenTox Algorithm and Model APIs are part of the OpenTox API that enables interaction among all OpenTox software components. The latest OpenTox API version is API 1.2 (www.opentox.org/dev/apis/api-1.2). Based on the REST principles mentioned before, each algorithm and each model, in RESTful terms, is a resource. Section 3 of this report presents the OpenTox algorithm and model API extensions to account for the special needs of ENM predictive toxicology and the fact that ENMs are often characterised by multitude of assays, resulting in high-dimensional datasets.

2) Development of Descriptors: For the development of predictive nQSAR models, the set of features is being extended beyond the classical chemical structure descriptors. Section 4 of this report describes the specifications for the tools that will be developed in this project for deriving descriptors from images, expressing the supramolecular pattern (size distribution, agglomeration state, shape, porosity and irregularity of the surface area) of ENMs as well as incorporating and grouping omics and high-content data as biological descriptors.
3) Modelling Algorithms and Methods: Section 5 describes how the OpenTox modelling resources will be extended to reflect the required adequate description of ENMs and integrate machine learning algorithms and validation tools, able to exploit and integrate diverse data and metadata captured in the database warehouse of the project including images, high throughput screening and omics data. Special emphasis will be given to the implementation of feature selection and clustering algorithms, which will allow investigating the causalities and associations between ENM characteristics and the interactions of ENM with biological systems, and grouping of ENMs into classes. The development of optimal experimental design facilities will support and advance the synergy between experimental and computational scientists in order to generate in a focused and efficient way reliable, consistent and rich-in-information experimental data.
4) Integration of R into the eNanoMapper computational infrastructure - Development of a QSAR modelling package in the R language: This work aims to integrate R/Bioconductor with the extended OpenTox based on the OpenCPU (Ooms, 2014) APIs. This will allow the integration of R statistical algorithms within the eNanoMapper computational infrastructure as well as the development of optimal nQSAR models by searching over many alternative modelling algorithms and tuning parameters. It will also facilitate the creation of Mechanism of Action QSARs by combining biological knowledge on mechanisms and pathways included in public ontologies and databases, such as Gene Ontology and KEGG.

3. OPENTOX API EXTENSIONS

The development of eNanoMapper on the OpenTox architecture requires extensions to be made to the OpenTox API in order to accommodate the particularities of ENMs in comparison to chemicals that drove the development of OpenTox APIs. The extensions that have been made in the functionality of Algorithms in comparison to the OpenTox API are focused on two main directions: firstly, introduction of PMML (Predictive Model Markup Language, 2014) for model definition and model description and secondly, modifications in Algorithm Options that allow users to perform optional actions on the dataset at the stage of model definition.

According to the OpenTox algorithm APIs, the representation of an algorithm contains information about the input a client should provide (obligatorily or optionally) to invoke the underlying procedure (e.g. training, data pre-processing etc.). The OpenTox POST algorithm and model services construct models and make predictions using datasets that are identified by unique URIs. In the original OpenTox algorithm API, there was the restriction that after identifying a URI as a prediction feature, the rest of the dataset should be used as input information. Quite often however, a user may wish to utilize only part of the input properties, replace missing values or perform transformations or scaling (normalization, standardization) on data. This is still possible with current OpenTox API, but through a sequence of calls to different web services, each one implementing a particular scaling or transformation procedure and producing a new dataset, which is stored in the database of the system. This workflow is not efficient, especially for large datasets, which are often produced by ENM toxicity studies, because it is time consuming and generates a number of intermediate datasets, which are of no value. The OpenTox algorithm APIs have been extended to allow going through all these pre-processing stages from a single POST call, so that transformations are calculated and used internally in the training procedure, without creating any additional features/properties and intermediate datasets. This became possible by using the PMML syntax, which is able to select a subset of properties (features) from the training dataset and also optionally apply transformations on these properties using an XML-based file. The extensions on OpenTox algorithm APIs are shown in Table 1.

The following optional parameters have been added in the POST method, as shown in Table 1:

- **normalization** is performed in the independent features/properties of the algorithm. If selected (normalization=1), it will be performed also in the prediction phase.
- **scaling** is performed in the independent features/properties of the algorithm. If selected (scaling=1), it will be performed also in the prediction phase. Default minimum and maximum scaling values are 0 and 1. It should be noted that scaling & normalization cannot be applied simultaneously.
- **mvh**. Missing Value Handling replaces missing values of a dataset. If selected (mvh=1), it is applied only in the training phase. It should be selected again in order to be performed in the prediction phase.
- **upload** is optional and refers to the PMML file that defines the features to be selected and the transformation to be applied on them.

Title	Method	URI	Parameters	Result	Supported MIME	Status codes
Get URIs of all available algorithms	GET	/algorithm	**[subjectid]** [?sameas=URI-of-the-owl:sameAs-entry]	List of all algorithm URIs or RDF representation, or algorithms of specific types, if query parameter exists Returns all algorithms, for which owl:sameAs is given by the query	text/uri-list, text/html	200, 404, 503
Get the ontology representation of an algorithm	GET	/algorithm/{id}	**[subjectid]**	Algorithm representation in one of the supported MIME types	text/uri-list, text/html, rdf/xml	200, 404, 503
Apply the algorithm	POST	/algorithm/{id}	**dataset_uri** **prediction_feature**, **parameter** (specified by the algorithm provider), **dataset_service**=datasetservice_uri, **result_dataset**, **[subjectid]** **scaling**=1, **scalingMin**=0, **scalingMax**=1 (optional), **normalization**=1 (optional), **mvh**=1 (optional), **upload**=file://path_to_pmml_file (optional)	*model URI (Prefer to create algorithm services that return model URIs instead of datasets or features)* *dataset URI* *featureURI*	text/uri-list, text/html, rdf/xml	200, 404, 503

Table 1 Extended OpenTox Algorithm API

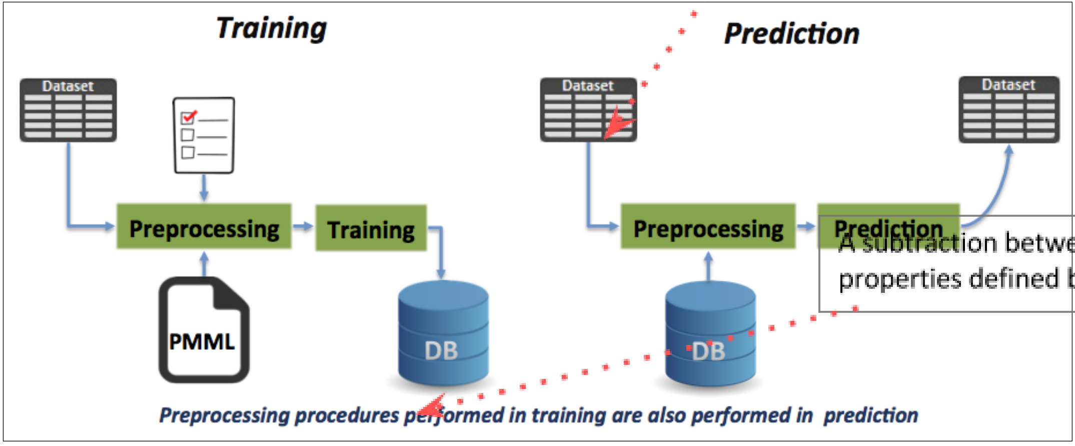
A typical example of an Input PMML file for the eNanoMapper web services follows next. Here, the dataset that will be used for training will contain two properties as independent properties. Their subtraction and their magnitude are calculated and used as descriptors during training, but no property URIs will be created for these internal descriptors.

```
<PMML version="4.0"
  xsi:schemaLocation="http://www.dmg.org/PMML-4_0
  http://www.dmg.org/v4-0/pmml-4-0.xsd"
  xmlns="http://www.dmg.org/PMML-4_0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<DataDictionary numberOfFields="4" >
 <DataField
  name="https://apps.ideaconsult.net/enmtest/property/P-
CHEM/ZETA_POTENTIAL_SECTION/ZETA+POTENTIAL/7F8B3FB82019B1CCF8A8C3FD2B5A2DACBDDDB832/3ed642f9-
1b42-387a-9966-dea5b91e5f8a"
  optype="continuous" dataType="double" />
 <DataField
  name="https://apps.ideaconsult.net/enmtest/property/P-
CHEM/ZETA_POTENTIAL_SECTION/ZETA+POTENTIAL/06399AE1609F65589E8D7C6DECF4A7E8565336CA/3ed642f9-1b42-
387a-9966-dea5b91e5f8a"
  optype="continuous" dataType="double" />
</DataDictionary>
<TransformationDictionary>
        <DerivedField dataType="double" name="zp_ch" optype="categorical">
                <Apply function="-">
                        <FieldRef field="https://apps.ideaconsult.net/enmtest/property/P-
CHEM/ZETA_POTENTIAL_SECTION/ZETA+POTENTIAL/7F8B3FB82019B1CCF8A8C3FD2B5A2DACBDDDB832/3ed642f9-
1b42-387a-9966-dea5b91e5f8a"/>
                        <FieldRef field="https://apps.ideaconsult.net/enmtest/property/P-
CHEM/ZETA_POTENTIAL_SECTION/ZETA+POTENTIAL/06399AE1609F65589E8D7C6DECF4A7E8565336CA/3ed642f9-1b42-
387a-9966-dea5b91e5f8a"/>
                </Apply>
        </DerivedField>
        <DerivedField dataType="double" name="zp_synth_mag" optype="categorical">
                <Apply function="abs">
                        <FieldRef field="https://apps.ideaconsult.net/enmtest/property/P-
CHEM/ZETA_POTENTIAL_SECTION/ZETA+POTENTIAL/7F8B3FB82019B1CCF8A8C3FD2B5A2DACBDDDB832/3ed642f9-
1b42-387a-9966-dea5b91e5f8a"/>
                </Apply>
        </DerivedField>
        <DerivedField dataType="double" name="zp_serum_mag" optype="categorical">
                <Apply function="abs">
                        <FieldRef field="https://apps.ideaconsult.net/enmtest/property/P-
CHEM/ZETA_POTENTIAL_SECTION/ZETA+POTENTIAL/06399AE1609F65589E8D7C6DECF4A7E8565336CA/3ed642f9-1b42-
387a-9966-dea5b91e5f8a"/>
                </Apply>
        </DerivedField>
</TransformationDictionary>
```

```
</PMML>
```

As shown in **Figure 1**, the PMML input file is provided together with the parameters and defines actions that are executed on the training dataset during pre-processing, then model training is performed and the model is stored in the database. As a next step, when the model is called from the database, the parameters and PMML actions used in the training stage will be applied to the dataset for which we want to make predictions and a final dataset is produced, that contains the values from the initial dataset augmented with predictions and information on the Domain of Applicability (DoA).

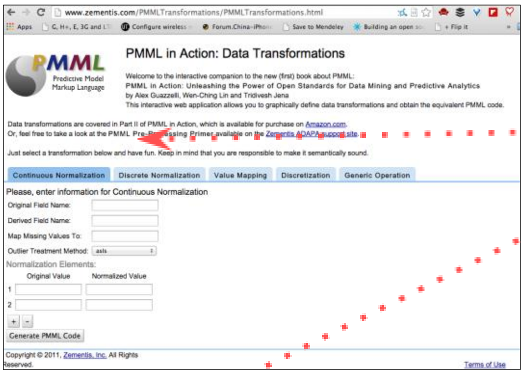
The URIs for the two properties that will be used are defined here



A subtraction between two properties defined below

Figure 1 Use of PMML files during Training and Prediction

A quick way to produce PMML code without actually writing the code (although actually writing the code is probably faster for the savvy) is available at the webpage (Figure 2): http://www.zementis.com/PMMLTransformations/PMMLTransformations.html. There, users can fill in the property URIs that will be used, choose the actions to be performed on them and the code is generated automatically.



PMML in Action: Data Transformations

Welcome to the interactive companion to the new (first) book about PMML:
PMML in Action: Unleashing the Power of Open Standards for Data Mining and Predictive Analytics
by Alex Guazzelli, Wen-Ching Lin and Tridivesh Jena
This interactive web application allows you to graphically define data transformations and obtain the equivalent PMML code.

Data transformations are covered in Part II of PMML in Action, which is available for purchase on Amazon.com.
Or, feel free to take a look at the PMML Pre-Processing Primer available on the Zementis ADAPA support site.

Just select a transformation below and have fun. Keep in mind that you are responsible to make it semantically sound.

Please, enter information for Continuous Normalization

Original Field Name:
Derived Field Name:
Map Missing Values To:
Outlier Treatment Method: asIs

Normalization Elements:

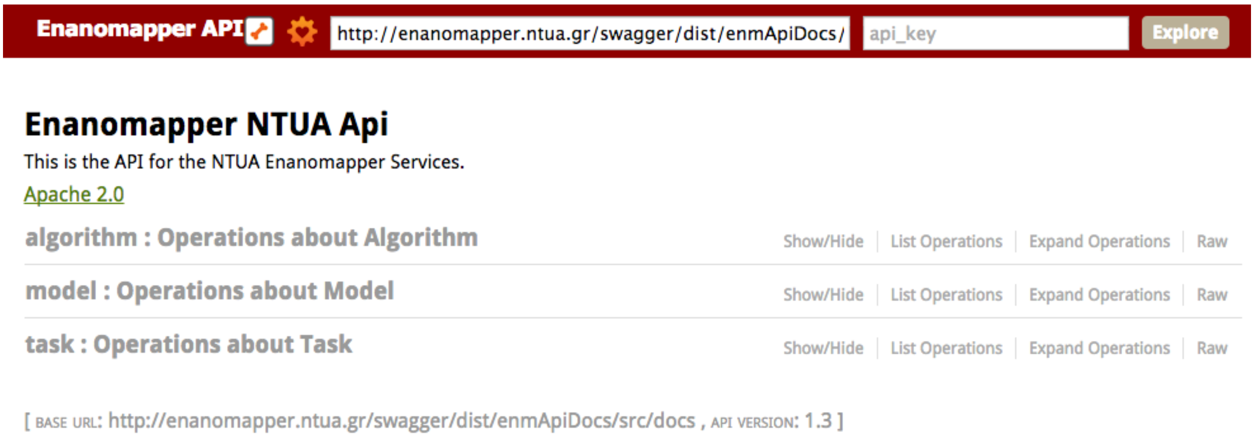
	Original Value	Normalized Value
1		
2		

Copyright © 2011, Zementis, Inc. All Rights Reserved.

The absolute value of the property below

Figure 2 Web tool that generates PMML code.

The algorithm and model REST API has been documented using Swagger (http://swagger.io/) and is available at http://enanomapper.ntua.gr/swagger/dist/ (Figure 3). As can be noticed in this figure, on the top right corner, a field named **api_key** is required. This is a token that identifies the user for all OpenTox web services and allows access to certain resources. For convenience, the examples provided here have been structured to be accessible by the **guest** account.



Enanomapper NTUA Api

This is the API for the NTUA Enanomapper Services.

Apache 2.0

algorithm : Operations about Algorithm — Show/Hide | List Operations | Expand Operations | Raw

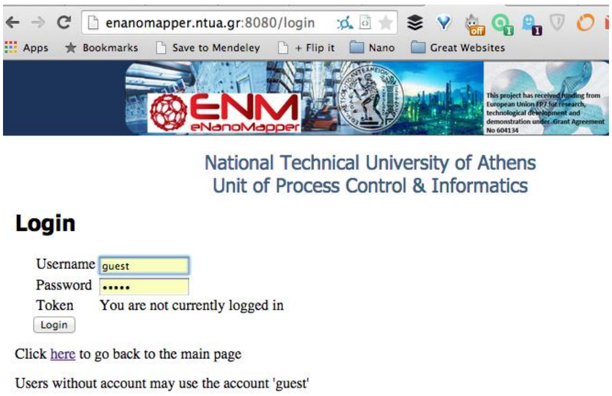
model : Operations about Model — Show/Hide | List Operations | Expand Operations | Raw

task : Operations about Task — Show/Hide | List Operations | Expand Operations | Raw

[BASE URL: http://enanomapper.ntua.gr/swagger/dist/enmApiDocs/src/docs , API VERSION: 1.3]

Figure 3 Swagger for algorithm, model and task.

The token can be provided to the user by visiting http://enanomapper.ntua.gr:8080/login (Figure 4) and filling in the Username and Password fields with “guest”. A string is provided (Figure 5) that the user should copy and paste in the **api_key** field of the Swagger interface (Figure 3).



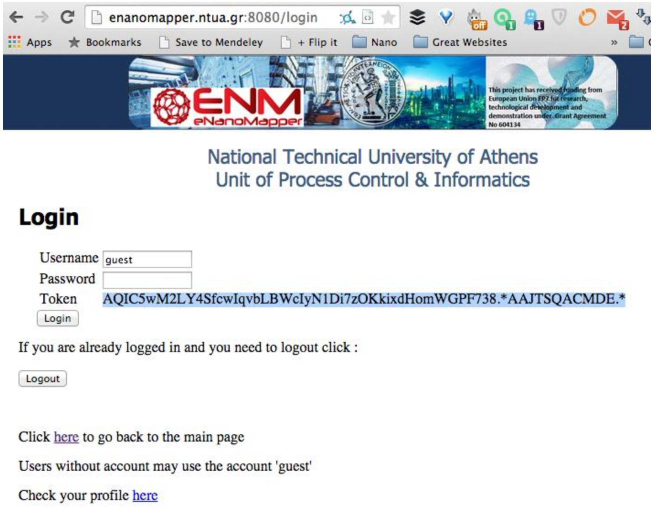
National Technical University of Athens
Unit of Process Control & Informatics

Login

Username guest
Password •••••
Token You are not currently logged in
Login

Click here to go back to the main page

Users without account may use the account 'guest'



National Technical University of Athens
Unit of Process Control & Informatics

Login

Username guest
Password
Token AQIC5wM2LY4SfcwIqvbLBWcIyN1Di7zOKkixdHomWGPF738.*AAJTSQACMDE.*
Login

If you are already logged in and you need to logout click :

Logout

Click here to go back to the main page

Users without account may use the account 'guest'

Check your profile here

Figure 4 Login page for NTUA web services

Figure 5 Token provided after login

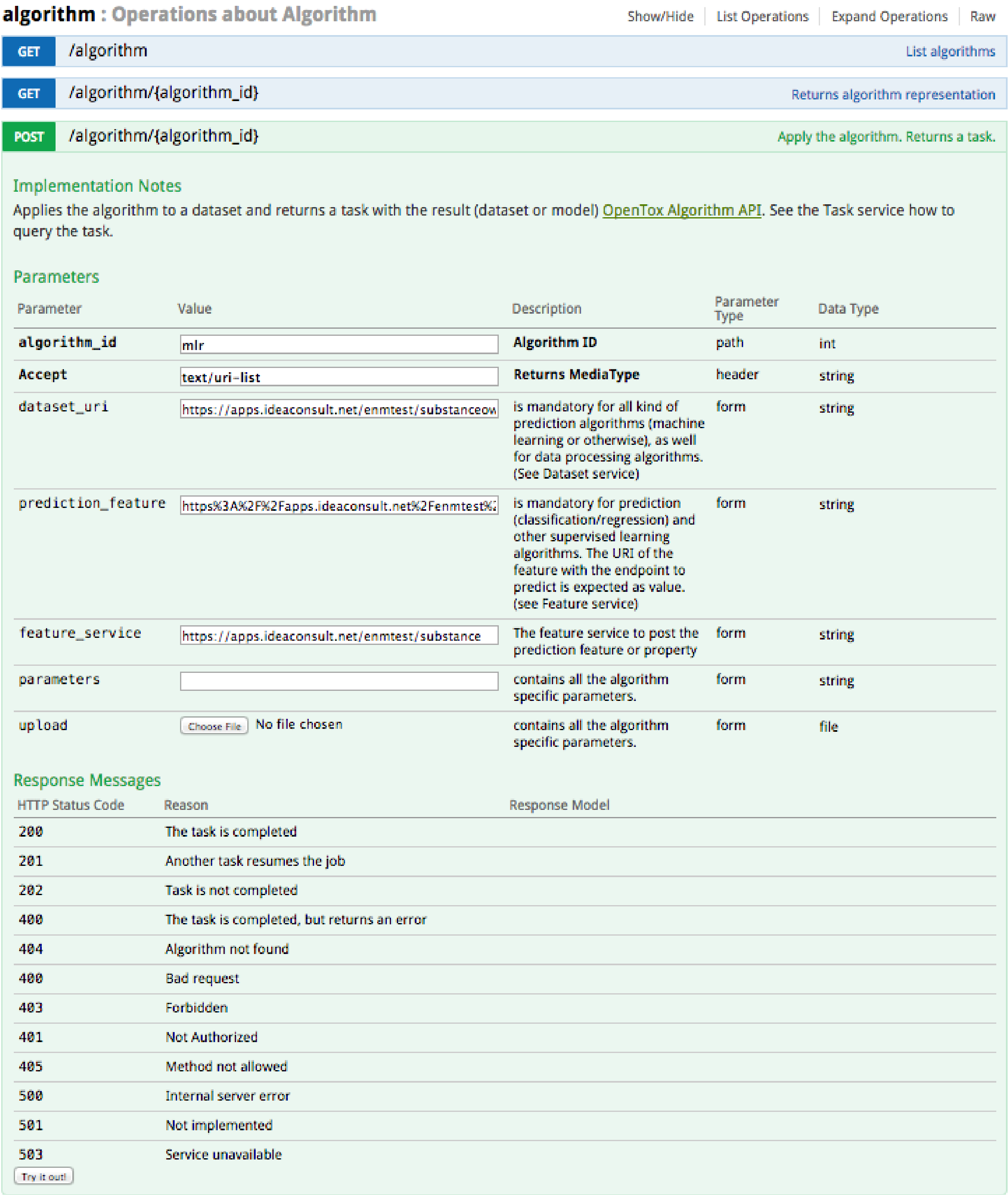
As an example the algorithm POST documentation will be explained in details. Selecting algorithm and then POST /algorithm/{algorithm_id} leads the user to the interface shown in Figure 6 that allows POSTing a dataset to the algorithm web service and seeing the web service in action. All the required fields have been prefilled, but the user needs to upload a PMML file to select the subset of variables that will be used to make the model.

The dataset URI: https://apps.ideaconsult.net/enmtest/substanceowner/FCSV-953F80D1-74C2-3127-B179-3AA4275D10B9/dataset points to the protein corona dataset that is described section 4.2 of deliverable 3.1

The PMML file described on page 13 is applicable here and is provided at this address: http://enanomapper.ntua.gr/enanomapper/images/0/01/JaqpotDemoPMML.xml. This particular PMML selects two properties, namely Zeta Potential measured in a solution with and without Human Serum (identified by their respective URIs https://apps.ideaconsult.net/enmtest/property/P-CHEM/ZETA_POTENTIAL_SECTION/ZETA+POTENTIAL/06399AE1609F65589E8D7C6DECF4A7E8565336CA/3ed642f9-1b42-387a-9966-dea5b91e5f8a and https://apps.ideaconsult.net/enmtest/property/P-CHEM/ZETA_POTENTIAL_SECTION/ZETA+POTENTIAL/7F8B3FB82019B1CCF8A8C3FD2B5A2DACBDDDB832/3ed642f9-1b42-387a-9966-dea5b91e5f8a) and passes 4 intermediate variables as input information to the model: their absolute values, difference and quotient. This gives the modeler the ability to look for insight that may be extracted from the difference in values encountered among measurements under different conditions.

Next is the syntax of a CURL command to the algorithm POST method that posts the same dataset to the MLR algorithm to predict the same feature (call to local PMML file is given in last line and refers to the same PMML as above).

```
curl -X POST http://enanomapper.ntua.gr:8080/algorithm/mlr -F
dataset_uri=https://apps.ideaconsult.net/enmtest/substanceowner/FCSV-953F80D1-74C2-3127-B179-
3AA4275D10B9/dataset -F
prediction_feature=https%3A%2F%2Fapps.ideaconsult.net%2Fenmtest%2Fproperty%2FTOX%2FUNKNOWN_TOXI
CITY_SECTION%2FLog2%2Btransformed%2F94D664CFE4929A0F400A5AD8CA733B52E049A688%2F3ed642f9-
1b42-387a-9966-dea5b91e5f8a -F feature_service=https://apps.ideaconsult.net/enmtest/substance -H Content-
type:multipart/form-data -H 'Accept:text/uri-list' -H 'subjectid:AQIC5wM2LY4SfczFbKg9zcO-
Uitq0d9wfYRd2yOftu6zc6Y.*AAJTSQACMDE.*' -F 'upload=@/Users/Philip/Downloads/JaqpotDemoPMML.xml' -i
```


algorithm : Operations about Algorithm
Show/Hide
List Operations
Expand Operations
Raw
GET
/algorithm
List algorithms
GET
/algorithm/{algorithm_id}
Returns algorithm representation
POST
/algorithm/{algorithm_id}
Apply the algorithm. Returns a task.
Try it out!

Implementation Notes

Applies the algorithm to a dataset and returns a task with the result (dataset or model) OpenTox Algorithm API. See the Task service how to query the task.

Parameters

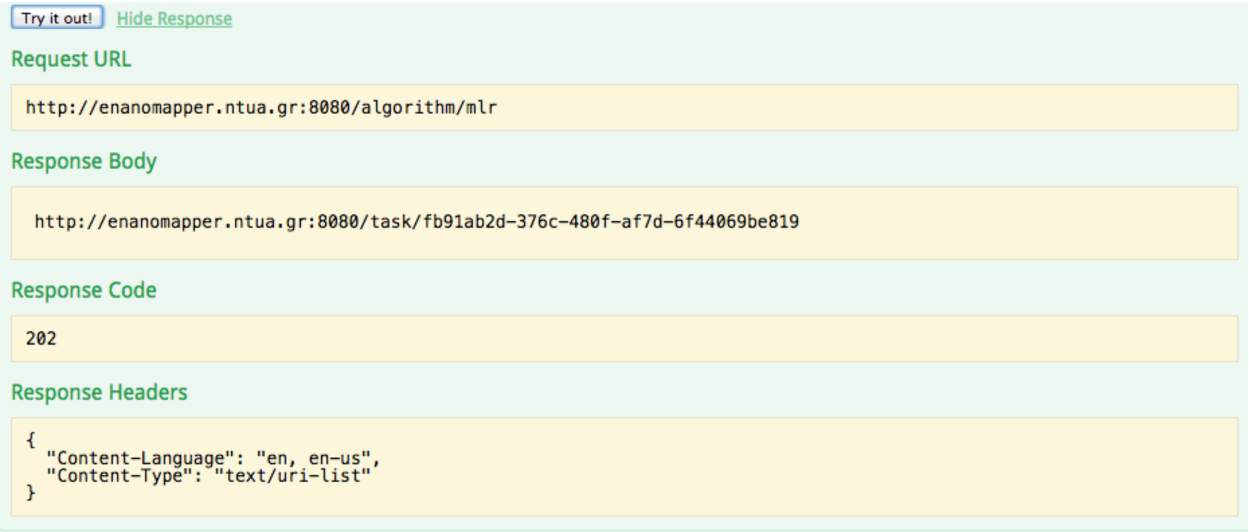
Parameter	Value	Description	Parameter Type	Data Type
algorithm_id	mlr	**Algorithm ID**	path	int
Accept	text/uri-list	**Returns MediaType**	header	string
dataset_uri	https://apps.ideaconsult.net/enmtest/substanceow	is mandatory for all kind of prediction algorithms (machine learning or otherwise), as well for data processing algorithms. (See Dataset service)	form	string
prediction_feature	https%3A%2F%2Fapps.ideaconsult.net%2Fenmtest%2	is mandatory for prediction (classification/regression) and other supervised learning algorithms. The URI of the feature with the endpoint to predict is expected as value. (see Feature service)	form	string
feature_service	https://apps.ideaconsult.net/enmtest/substance	The feature service to post the prediction feature or property	form	string
parameters		contains all the algorithm specific parameters.	form	string
upload	Choose File No file chosen	contains all the algorithm specific parameters.	form	file

Response Messages

HTTP Status Code	Reason	Response Model
200	The task is completed	
201	Another task resumes the job	
202	Task is not completed	
400	The task is completed, but returns an error	
404	Algorithm not found	
400	Bad request	
403	Forbidden	
401	Not Authorized	
405	Method not allowed	
500	Internal server error	
501	Not implemented	
503	Service unavailable	

Figure 6 Algorithm POST operation API documentation

After POSTing the dataset by clicking the **Try it out!** button, the page expands to provide the response from Swagger (Figure 7), which is a Task URI that can be accessed on a web browser and gives information on the status of the Task that was initiated (Figure 8). In that page, the URI of the model that was created is provided (named as Result URI) to be used to make predictions.



Request URL

```
http://enanomapper.ntua.gr:8080/algorithm/mlr
```

Response Body

```
http://enanomapper.ntua.gr:8080/task/fb91ab2d-376c-480f-af7d-6f44069be819
```

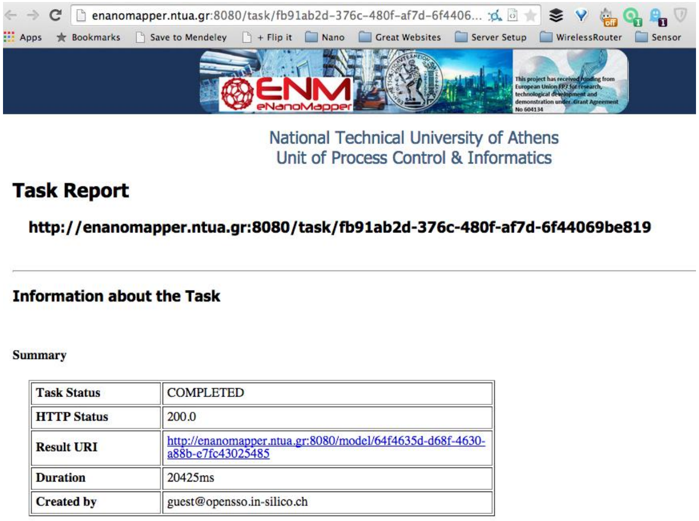
Response Code

```
202
```

Response Headers

```
{
  "Content-Language": "en, en-us",
  "Content-Type": "text/uri-list"
}
```

Figure 7 Response from trying algorithm POST operation


enanomapper.ntua.gr:8080/task/fb91ab2d-376c-480f-af7d-6f4406...
Apps
Bookmarks
Save to Mendeley
+ Flip it
Nano
Great Websites
Server Setup
WirelessRouter
Sensor
eNanoMapper
This project has received funding from European Union FP7 for research, technological development and demonstration under Grant Agreement No 604134

National Technical University of Athens
Unit of Process Control & Informatics

Task Report

http://enanomapper.ntua.gr:8080/task/fb91ab2d-376c-480f-af7d-6f44069be819

Information about the Task

Summary

Task Status	COMPLETED
HTTP Status	200.0
Result URI	http://enanomapper.ntua.gr:8080/model/64f4635d-d68f-4630-a88b-e7fc43025485
Duration	20425ms
Created by	guest@opensso.in-silico.ch

Figure 8 Task information and model URI

After creating models, transparency and ease of transfer are two important factors in the acceptance of models by the community and propagation within it. As an extension over the OpenTox algorithm API, the produced model should be available in PMML format. This allows users to inspect the model at hand and possibly compare it to other models in literature or in other platforms, while at a second stage it can be transferred easily to other systems for comparison, validation or for final deployment. Examples of Multiple Linear Regression (MLR) and Radial Basis Function (RBF) models in PMML format are provided below.

MLR model in PMML format:

```
<PMML version="3.2">
<Model ID="a5a99d6e-44a3-43a8-8de1-f40151d0dc7b" Name="MLR Model">
<AlgorithmID href="http://localhost:8080//algorithm/mlr"/>
```

```
<DatasetID href="http://apps.ideaconsult.net:8080/ambit2/dataset/R545"/>
 <AlgorithmParameters /><Timestamp>Thu Aug 28 18:32:46 EEST 2014</Timestamp>
 </Model>
 <DataDictionary numberOfFields="4" >
                <DataField name="http://apps.ideaconsult.net:8080/ambit2/feature/22127" optype="continuous"
dataType="double"/>     <DataField name="http://apps.ideaconsult.net:8080/ambit2/feature/22252" optype="continuous"
dataType="double"/> </DataDictionary>
 <RegressionModel modelName="a5a99d6e-44a3-43a8-8de1-f40151d0dc7b" functionName="regression"
                                                 modelType="linearRegression" algorithmName="linearRegression">
                <MiningSchema>
                                <MiningField name="http://apps.ideaconsult.net:8080/ambit2/feature/22127"/>
                                <MiningField name="http://apps.ideaconsult.net:8080/ambit2/feature/22252"/>
                                <MiningField name="http://apps.ideaconsult.net:8080/ambit2/feature/22200"
usageType="predicted"/>                 </MiningSchema>
                <RegressionTable intercept="-5.1959636547035775">
                                <NumericPredictor name="http://apps.ideaconsult.net:8080/ambit2/feature/22127"
                                <NumericPredictor name="http://apps.ideaconsult.net:8080/ambit2/feature/22252"
                </RegressionTable>
      <ModelExplanation>
               <PredictiveModelQuality dataName="http://apps.ideaconsult.net:8080/ambit2/dataset/R545"
                                                             dataUsage="training" meanAbsoluteError="0.002"
                                                             rootMeanSquaredError="0.0024"/>
      </ModelExplanation>
 </RegressionModel>
 </PMML>
```

RBF model in PMML format:

```
<PMML version="3.2" xmlns="http://www.dmg.org/PMML-3_2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
<Header copyright="Copyleft (c) OpenTox - An OpenSource Predictive Toxicology Framework, http://www.opentox.org,
2009" />
<Model ID="c6a9b48f-1309-4a03-bc60-4caf995e4ca2" Name="fastRbfNn Model">
<AlgorithmID href="http://enanomapper.ntua.gr:8080/algorithm/fastRbfNn"/>
<DatasetID href="http://apps.ideaconsult.net:8080/ambit2/dataset/R545"/>
<AlgorithmParameters /> <Timestamp>Mon Sep 08 12:42:40 EEST 2014</Timestamp>
</Model>
<DataDictionary numberOfFields="4" >
                <DataField name="http://apps.ideaconsult.net:8080/ambit2/feature/22127" optype="continuous"
dataType="double" />    <DataField name="http://apps.ideaconsult.net:8080/ambit2/feature/22137" optype="continuous"
dataType="double" />    <DataField name="http://apps.ideaconsult.net:8080/ambit2/feature/22213" optype="continuous"
dataType="double" />    <DataField name="http://apps.ideaconsult.net:8080/ambit2/feature/22252" optype="continuous"
dataType="double" /> </DataDictionary>
<NeuralNetwork modelName="c6a9b48f-1309-4a03-bc60-4caf995e4ca2" functionName="regression"
activationFunction="radialBasis">       <MiningSchema>
                                <MiningField name="http://apps.ideaconsult.net:8080/ambit2/feature/22127" />
                                <MiningField name="http://apps.ideaconsult.net:8080/ambit2/feature/22137" />
                                <MiningField name="http://apps.ideaconsult.net:8080/ambit2/feature/22213" />
                                <MiningField name="http://apps.ideaconsult.net:8080/ambit2/feature/22252" />
                                <MiningField name="http://apps.ideaconsult.net:8080/ambit2/feature/22200"
usageType="predicted"/>                 </MiningSchema>
                <NeuralLayer numberOfNeurons="2">
                                <Neuron width="0.6364905064868956" id="0">
                                <Con from="0" weight="0.24735449254512787"/>
                                <Con from="1" weight="0.7419981956481934"/>
```

```
                    <Con from="2" weight="1.0"/>
                    <Con from="3" weight="0.3026672303676605"/>
                    </Neuron>
                    <Neuron width="1.272011433264412" id="1">
                    <Con from="0" weight="0.21022428572177887"/>
                    <Con from="1" weight="0.6841791272163391"/>
                    <Con from="2" weight="3.0"/>
                    <Con from="3" weight="0.3536739647388458"/>
                    </Neuron>  </NeuralLayer>
          <NeuralLayer numberOfNeurons="1">
                    <Neuron id="2">
                    <Con from="4" weight="-5.370853122331072"/>
                    <Con from="5" weight="-6.900625708161085"/>
                    </Neuron>
          </NeuralLayer>
          <NeuralOutputs numberOfOutputs="1">
                    <NeuralOutput outputNeuron="2">
                               <DerivedField optype="continuous" dataType="double"></DerivedField>
                          </NeuralOutput>
          </NeuralOutputs>
 </NeuralNetwork>
</PMML>
```

4. DESCRIPTOR DEVELOPMENT AND CALCULATIONS

4.1 QUANTUM MECHANICAL DESCRIPTORS FOR NANOMATERIALS

Quantum Mechanics allows calculations for descriptors of ENMs related to their electronic structure. There is a lot of specialised software for Quantum Mechanical calculations that uses methodologies previously applied to conventional materials and is considered compatible with ENMs with the proper parameters/modifications. The available software for descriptor calculations was studied systematically, in order to assess the available options. An index for the most relevant software that is used in the scientific community has been compiled and is given in Table 6 in the Appendix. Each package includes one or more methodologies and provides values for a group of available descriptors.

The development of Ab initio and first principle models with specialized software such as Quantum Espresso, GAMESS or SIESTA is time consuming, can be handled only be expert users and requires a lot of interactions with the software. Users of such software require a high level of access to intermediate results while the program is run, in order to scrutinize the calculations, something that is best achieved by them running the calculations on their own equipment. Therefore, development of services automating the generation of quantum mechanical descriptors using such software will be inefficient and not beneficial and practical for the purposes of eNanoMapper. The users should still be allowed to upload results produced by such software.

Semi-empirical methods like PM6 or PM7 are much faster and have already been used for describing metal oxide ENMs (Puzyn et al., 2011). The current eNanoMapper MOPAC implementation (https://apps.ideaconsult.net/enanomapper/algorithm/ambit2.mopac.MopacShell) has been tested thoroughly on metal oxides and the results are successful when the metal ions are used. As an example, the following link:

http://apps.ideaconsult.net:8080/enmtest/ui/_dataset?dataset_uri=http%3A%2F%2Fapps.ideaconsult.net%3A8080%2Fenmtest%2Fdataset%2F11%3Ffeature_uris%5B%5D%3Dhttp%253A%252F%252Fapps.ideaconsult.net%253A8080%252Fenmtest%252Fmodel%252F5%252Fpredicted

contains quantum mechanical descriptors calculated using the OpenTox MOPAC web service for Sb_2O_3 , which is represented by the following URI:

http://apps.ideaconsult.net:8080/enmtest/dataset/11

The following table contains MOPAC calculations of the Enthalpy of formation of a gaseous action for several metal oxides using the PM6 and PM7 methods and comparison with the results available in the literature (Puzyn et al., 2011):

	Bibliography	PM6	PM7
Al2O3	1187,83	1187,83	1512,06
Bi2O3	1137,4	1137,40	1177,37
CoO	601,8	594,58	688,16
Cr2O3	1268,7	1266,40	1679,51
Fe2O3	1408,29	1363,40	1476,01
In2O3	1271,13	1271,13	1275,37
NiO	596,7	596,70	601,30
Sb2O3	1233,06	1233,06	1099,49
SiO2	1686,38	1686,38	2344,98
SnO2	1717,32	1717,32	2078,46
TiO2	1575,73	1575,73	1838,52
V2O3	1097,73	1097,73	1156,56
Y2O3	*837,15*	*837,15*	*852,66*
ZnO	662,44	662,44	653,31
ZrO2	1357,66	1357,66	1391,47
CuO	706,25	713,74	574,08

Table 2: Enthalpy of formation of a gaseous cation: Bibliography vs. MOPAC calculations for metal oxide ENMs (literature data from Puzyn 2011)

Semi-empirical methods can still be used, when the complete ENM structure needs to be introduced for calculating QM descriptors. In order to perform such calculations, we downloaded Crystallographic Information Files (Brown and McMahon, 2002) (with the extension .CIF) available publically (Crystallography Open Database http://www.crystallography.net/ and sources found during web search) for describing the 3-D crystallographic structure of metal oxide ENMs, which however needed to be transformed into files that could be used as input to the MOPAC software. Different software like OpenBabel, Mercury and CIFtr were employed to draw out a workflow that allowed file format conversion while retaining the structure information. The results for HOMO and LUMO for Y_2O_3 are shown in Table 3, they reveal however that there is some inconsistency among the results produced by different QM descriptor calculation algorithms and options.

HOMO eV	LUMO eV	GAP	Source	Parameters
-1,28	1,2	-2,48	Puzyn 2011, PM6	-
-4,05	-2,65	-1,4	Hussain 2012, PM6	-
-2,86	-0,62	-2,24	*PM6*	with geometry optimization
-3,65	-1,46	-2,19	*PM6*	without geometry optimization

Table 3: Y_2O_3: Comparison between literature data and calculations for (by method and parameters used).

4.2 IMAGE DERIVED DESCRIPTORS

Transmission electron microscopy (TEM) is an important ENM characterization technique. TEM image analysis yields number-based results, allows size and specific shape measurements and characterization of surface topologies, and provides distinctions between the characterizations of primary particles and of aggregates/agglomerates. Based on TEM images, Puzyn et. al. (2011) have proposed a set of image derived descriptors for characterizing ENMs that are summarized in Table 4. This will be the minimum set of descriptors to be computed by an image analysis tool that will be developed in the context of the eNanoMapper project, based on the standard and well accepted Fiji/ImageJ (Abràmoff et.al., 2004) open-source software. Fiji/ImageJ was selected after an assessment of the most relevant software tools that are available and in use by the scientific community, which are given in *Table* 7 in the Appendix.

	Name	Definition of image descriptor	
Descriptors reflecting nanoparticle size	Volume (V)	The sum of all non-zero pixels episodes, combining measures of contour elements, assuming that the contour is a square with a side equal to the unity	
	Surface diameter (d_S)	The diameter of a sphere having the same surface area as the projected particle	$□_□ = \sqrt{\frac{□}{□}}$
	Equivalent volume diameter (d_V)	The diameter of a sphere having the same volume as the projected particle	$□_{□_□} = \sqrt[3]{\frac{6□}{□}}$
	Equivalent volume/surface (d_{Sauter})	The diameter of a sphere having the same volume to surface ratio as the projected particle	$□_{□□□□□□} = \frac{6□}{□}$
Descriptors reflecting nanoparticle surface area	Area (A)	The sum of the all non-zero pixels (x_i)	$□ = \sum_{□=1}^{□} □_□$
	Porosity (P_x)	The sum of the relative differences in intensities between values of neighbouring pixels (x_i and y_i) along the X axis	$□_□ = \sum_{□=1}^{□} \lvert □_□ - □_□ \rvert$
	Porosity (P_y)	The sum of the relative differences in intensities between values of neighbouring pixels (x_i and y_i) along the Y axis	$□_□ = \sum_{□=1}^{□} \lvert □_□ - □_□ \rvert$
Descriptors reflecting nanoparticle shape	Sphericity (Ψ)	The ratio of the surface area of a sphere - with the same volume as the particle considered - to the surface area of the particle	$□ = \frac{□^{\frac{1}{3}} 6□^{\frac{2}{3}}}{□}$
	Circularity (f_{circ})	The function of the surface area of the particle (A) and the particle's perimeter (V^2)	$□_{□□□□} = \frac{4□□}{□^{2}}$
	Anisotropy ratio (AR_X)	The ratio of the minimum length of chord of the X axis and the maximum length of chord of the Y axis.	$□_{□\_□} = \frac{□_{□□□\_□}}{□_{□□□\_□}}$
	Anisotropy ratio (AR_Y)	The ratio of the minimum length of chord of the Y axis and the maximum length of chord	$□_{□\_□} = \frac{□_{□□□\_□}}{□_{□□□\_□}}$

		of the X axis.	

Table 4: Image descriptors, definition and formula

Using this tool, the user will enter a web page hosted in a local machine and will be able to upload one or multiple images for analysis. After the upload procedure is done, he will select from a drop-down list the type of filter he wants to apply to the image and at the same time a preview of the image will be displayed on the screen. Then, the user will select from a checklist the descriptors that should be calculated and submit the form to the server. The server will run the descriptor calculation algorithms and the results will be returned in various forms.

The main features of the image analysis tool are:

- File manager to upload image
- Checklist with all the descriptors that can be calculated in the application (some of them will be preselected)
- Checklist with the filters a user can apply on the image. Among other, the following filters will be available: Huang, Intermodes, IsoData, MaxEntropy, Mean, MinError(I), Tri, Yen
- The result of the calculations will be saved in Excel, CSV or JSON format
- The pictures should not be ultra high definition. In order to avoid overkill procedures in the server the density and resolution of pictures should be at most full HD (1920*1080). This resolution is currently discussable according to the machine we are going to use and the request frequency.

Implementation of the image descriptor calculation tool is in progress and uses the following tools: JDK 1.7, Maven 4.01, ImageJ 1.47, JUnit 4, JSF 2 (primefaces). The source code is available at the following link: https://github.com/enanomapper/imageAnalysis. The service is under development and its current implementation can be found in the following address: http://enanomapper.ntua.gr:8880/imageAnalysis/ . A detailed description of this tool will be included in D4.2.

4.3 DESCRIPTORS DERIVED BY OMICS DATA

Recent studies have shown that the presence of serum proteins within *in vitro* cell culture systems forms a protein adsorption layer (a.k.a. the "protein corona") on the surface of nanoparticles that affects nanoparticle-cell interactions and cell response (Ge et al., 2011; Lesniak et al., 2012). The protein corona establishes nanoparticle's 'biological identity', i.e. nanoparticle's view as understood by the components of the biological system it belongs (Walkey et al., 2014). The protein corona thus encodes information about the interface formed between the nanoparticle and the cell surface within a physiological environment. As an example, the protein corona dataset (Walkey et al., 2014), which has been used for testing the modelling services as described in section 3, contains published analysis results based on a comprehensive quantitative characterization of blood protein corona. Good correlations of protein corona with cell association –which was chosen as the model biological interaction end-point because of its relevance to inflammatory responses, biodistribution, and toxicity *in vivo* (Lesniak et al., 2012)- suggests that protein corona fingerprinting may be developed into a general strategy to predict the interaction of nanoparticles with biological systems.

Furthermore, high-throughput experimental methods are now becoming more routine, even for ENMs, and produce a wealth of omics data. Experimental methods include RNA deep sequencing transcriptomic approaches, oligonucleotide microarrays, mass spectrometry (MS) experiments, as well as additional knowledge on, for example, DNA methylation and microRNA regulation. The reason for this acceleration in multi-omics data production is that emerging evidence suggests that only a single data type is insufficient for understanding complex machinery, such as tumour behaviour (Chin and Gary, 2008), or even actual cellular responses to different conditions are best explained mechanistically when taking all omics levels into account. One of the main problems in the field of omics data analysis and 'Big Data' analytics remains the lack of a systematic approach for integrating various data, although many noteworthy efforts have been recently published, especially in cancer studies (Kim et al., 2014; Sass et al., 2013). A considerable part of them focuses on how they can simultaneously assess the biological meaning of the analysis outcome (Sass et al., 2013). For example, studies incorporating genomic knowledge such as pathways or protein–protein interaction networks based on transcriptomics and proteomics data, have been developed to increase their power in predicting biologically relevant information (Abraham et al., 2010; Yang et al., 2012; Balbin et al., 2013). The findings of those studies suggest that integrating omics data with genomic knowledge to construct pre-defined features, results in higher performance in predicting clinical outcomes and higher consistency between the results of different studies.

We are interested in producing a new set of descriptors that would efficiently summarize omics data and potentially include additional information. Our goal is to enrich the data using gene set information whilst emphasizing the importance of -omics data in modelling ENM toxicity. There are now over 300 web resources (see http://pathguide.org/) providing access to many thousands of pathways and networks that document millions of interactions between proteins, genes and small molecules. Amongst them Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG) are quite popular. GO (http://geneontology.org/) is a major bioinformatics initiative to annotate gene and gene products and maintain their controlled vocabulary. Each GO term (GO id) corresponds to a particular molecular function, biological process or cellular component (sub-domain areas); those sets of actions are performed by a set of genes in the cell. KEGG (http://www.genome.jp/kegg/) is a database resource for understanding high-level functions and utilities of the biological system, such as the cell, the organism and the ecosystem, from molecular-level information (Bioportal, BioPax). Another popular repository is the MSigDB database (Molecular Signatures Database, http://www.broadinstitute.org/gsea/msigdb/index.jsp) which is a collection of annotated gene sets that can be used for Gene Set Enrichment Analysis (GSEA) (Subramanian et al., 2005). Part of MSigDB covers GO and KEGG information, but other gene sets taken from the literature are also included.

GO was selected as the gold standard for annotation in three ontology branches, namely Cellular Components (CC), Molecular Functions (MF), and Biological Processes (BP) containing 41,694 classes. For the protein corona case, GO information specific to each protein corona will be used to calculate a new set of descriptors, referred to as GO descriptors. GO descriptors will focus on sets of proteins with common biological information, will summarize proteomics data to predict the biological responses to nanoparticles and estimate possible nano-bio interactions. More specifically, we intend to cluster the aggregated proteomics data with their relevant GO information, using hierarchical clustering. We will examine different cuts of the produced dendrogram in order to estimate groups of proteins. Those protein groups will be summarized to GO descriptors by averaging their values. The final set of GO descriptors selected by nanoQSAR models will be further exploited for their biological relevance and functional similarity.

5. MODELLING ALGORITHMS AND METHODS

Through the OpenTox project a vast number of statistical methods or state of the art machine learning algorithms have been implemented as REST web services, including all major classification, regression and clustering algorithms from the generic Weka machine learning library (Hall, 2009). Indicative algorithms are decision tree algorithms, simple linear regression, multiple linear regression, support vector machines, nearest neighbour methods (kNN), complex machine learning algorithms, such as Bayes Net, as well as specialized learning algorithms developed by consortium partners, such as lazar which automates the read-across procedure (Helma, 2006). A number of feature selection, data transformation, dimensionality reduction and applicability domain estimation algorithms (including, but not limited to, partial least-squares filter, principle components analysis, chi-squared attribute evaluation, information-gain attribute evaluation) have also been implemented. Algorithm and model validation is supported by a dedicated validation service (testing on validation sets, bootstrapping, cross-validation, and comparison of validation results). The complete list of available OpenTox algorithms and methods can be found in the following link:
http://www.opentox.org/dev/documentation/components/ .

With the modification and extensions of the OpenTox API all supported state-of-the-art statistical and machine learning methods can be applied also to ENMs. OpenTox extension of modelling infrastructure will provide additional facilities for data analysis, and for building and validation of new ENM predictive models, which may be applied to all new datasets incorporated into the eNanoMapper infrastructure. It will also open interesting possibilities for exploring novel nanoparticle descriptors and similarity indices based e.g. on physicochemical parameters, image analysis, gene or protein expression responses or pathway analysis (Nel, 2012), leading ultimately to a better understanding of the key factors influencing nanotoxicity and to guidelines for the design of safer ENMs. Integration of the facilities provided by the R statistical language, (as described in the next section of the deliverable) will allow easy access to a wealth of additional algorithms and methods focusing on the analysis of omics data and utilization of useful information included in public ontologies such as the GO and KEGG ontologies. The extended OpenTox infrastructure will also provide the means for optimal experimental design through the development and incorporation of suitable algorithms and computational tools.

In the following section we present algorithms that have been already integrated or are planned to be utilized within the eNanoMapper computational infrastructure.

5.1 REGRESSION AND CLASSIFICATION ALGORITHMS

Least Absolute Shrinkage and Selection Operator (LASSO)

The LASSO (Tibshirani, 1996) is a regression method similar to Ordinary Least Squares (OLS) regression. LASSO minimizes the Residual Sum of Squares (RSS) but poses a constraint to the sum of the absolute values of the coefficients being less than a constant. This additional constraint is moreover similar to that introduced in Ridge regression, where the constraint is to the sum of the squared values of the coefficients. This simple modification allows LASSO to perform also variable selection because the shrinkage of the coefficients is such that some coefficients can be shrunk exactly to zero. It can be said that LASSO is an improvement over Ridge, in that LASSO has the beneficial aspects of Ridge, *i.e.* higher bias and lower variance (compared to OLS), but also allows to select variables, leading to an enhanced interpretability of the developed models.

Lasso is an eager learning algorithm. The tuning parameter of the model is the fraction, the L1 norm of the coefficient vector, which can, for example, be set to vary in a sequence of 10 values between zero and one.

Elastic Net

The Elastic Net is a regression method (Zou and Hastie, 2005) that combines the penalty terms of LASSO and Ridge regression. The Ridge term allows to shrink the coefficients, whereas the LASSO term is able to shrink some coefficients to 0, thus performing variable selection. The two terms can be properly tuned by an extra parameter (α*)*, depending on the problem under analysis. The Elastic Net method seems to be particularly useful when dealing with highly correlated variables. In such a situation the Ridge term shrinks coefficients of correlated variables toward each other, whereas the LASSO term picks one among the correlated variables and puts all weight on it.

Elastic Net is an eager learning algorithm. The tuning parameters of the model are called alpha and lambda. Alpha is the Elastic Net mixing parameter, taking values in [0,1] where □ = *1* corresponds to the LASSO penalty and □ = *0* corresponds to the Ridge penalty. Lambda is a regularization parameter, typically a sequence of values with the first value being very close to zero.

Random Forest

Random forests (RF) are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest (Bremen, 2001). RF is an ensemble approach and the decision trees generated can be used for classification. Each classifier individually is a "weak learner", while all classifiers together are a "strong learner". A standard decision tree corresponds to a "weak learner", where in RF each node in the tree is split using the best among a subset of predictors randomly chosen at that node. This modelling algorithm includes only two parameters (the number of variables in the random subset at each node and the number of trees in the forest). RF is an eager learning algorithm.

5.2 Clustering Algorithms

Hierarchical Clustering

Hierarchical clustering (HC) is a typical clustering analysis approach, which partitions the data sequentially, and outputs a hierarchy between clusters. In HC the goal is to construct nested partition layer by layer via grouping objects into a tree of clusters (Kaufman and Rousseau, 1990). A distance matrix is used as clustering criteria, which is typically computed using the Euclidean distance, Manhattan distance, Minkowski distance etc. A diagram, called dendrogram, graphically represents this hierarchy and is an inverted tree that describes the order in which objects are merged (bottom-up view) or clusters are split (top-down view). Subtrees in the dendrogram are ordered in such a way that the tighter cluster is on the left (the last, i.e., most recent, merge of the left subtree is at a lower value than the last merge of the right subtree). Single observations are the tightest clusters possible, and merges involving two observations place them in order by their observation sequence number.

HC are eager learning algorithms, and most of them are deterministic. Hierarchical techniques do not assume any particular number of clusters.

There are two basic methods to generating HC:

i. **Agglomerative**: Each object belongs to an individual cluster (singleton) and, at each step, the most similar pair of clusters are merged.
ii. **Divisive:** All objects belong to one cluster and, at each step, a cluster is split until only singleton clusters of individual objects remain.

Typical clustering methods are:

- Single-link (or single-linkage) clustering which is a local criterion merging clusters that are the closest to each other,
- Complete-link (or complete-linkage) clustering which is a non-local criterion where the similarity of the two clusters is the similarity of their most dissimilar members,
- Average-link clustering which computes the average similarity of all pairs in the two clusters including pairs from the same cluster,
- Ward’s minimum variance method which aims at finding compact, spherical clusters.

Other methods, such as median, centroid, McQuity, are aiming for clusters with characteristics somewhere between the single- and complete- link methods (Legendre and Legendre, 2012).

Bi-clustering

A more refined, local, approach for clustering is known as biclustering. While clustering methods can be applied to either rows or columns of a data matrix separately, biclustering methods perform clustering in the two dimensions simultaneously (Cheng and Church, 2000; Lazzeroni and Owen, 2000). Its basic difference with other clustering analysis is that while clusters always create disjoint clusters that cover all the input set, biclusters may overlap, and they usually cover only a part of the matrix. This overlap is expected when assuming that each bicluster represents a function, or quality of the data. Biclustering takes as an input a distance matrix and tries to find statistically significant sub-matrices in it, also called biclusters. These structures imply a joint behaviour of some objects under some conditions.

Biclustering is an eager learning algorithm. Biclustering is able to co-cluster binary, contingency, continuous and categorical data, applying an Expectation-Maximization algorithm and offering an option for semi-supervised co-clustering. The user can specify the number of classes in both directions, or declare them as unknown.

Supervised Clustering

The methodologies mentioned above belong to the family of unsupervised learning techniques, i.e. they estimate hidden structure in unlabelled data. In the case of supervised clustering, label information of the data is known, i.e. we assume that samples are classified. The goal of supervised clustering is to identify class-uniform clusters that have high probability densities (Aggarwal et al., 1999; Bansal et al., 2002). Moreover, in supervised clustering, objects are assigned to clusters using a notion of similarity with respect to a given distance function.

5.3 Feature Selection Algorithms

Variable Importance on Partial Least Squares projections (VIP)

VIP is a variable selection method based on the Canonical Powered PLS (CPPLS) regression (Indahl et al., 2009). The CPPLS algorithm assumes that the column space of **X** (independent variables) has a subspace of dimension M containing all information relevant for predicting the dependent variable. This subspace is known as the relevant subspace. The different strategies for PLS-based variable selection are usually based on a rotation of the standard solution by a manipulation of the PLS weight vector (**w**) or the regression coefficient vector, **b**. The VIP method selects variables by calculating the VIP score for each variable and excluding all the variables with VIP score below a predefined threshold *u*. All the parameters that provide an increase in the predictive ability of the model are retained (Indahl et al., 2009). The “greater than one” rule (u = 1) is generally used as a criterion for variable selection because the average of squared *VIP* scores is equal to 1.

Recursive Feature Elimination

Recursive feature elimination (RFE) is a backwards variable selection method. The RFE algorithm fits the model to all predictors, where each predictor is ranked using its importance to the model. At each iteration of the feature selection, the S top ranked predictors are retained, the model is refit and performance is assessed. The predictor rankings could be recomputed on each reduced feature set, which would generally increase performance, although it has been shown that in some algorithms (e.g. Random Forest) there is a decrease in performance (Svetnik et al., 2004).

Tuning parameters include the specification of the number of features that should be retained, and the external resampling method used (options are bootstrap, LOOCV, CV, repeated LOOCV).

Genetic Algorithms

Selected and unselected features are represented by binary sequences of 1s and 0s (chromosomes) where the value of 1 means that the feature is selected, while the value of 0 corresponds to an unselected feature. In each iteration of the main loop, chromosomes are selected using the “roulette selection” method. In this way, the best chromosomes tend to be selected more frequently and the worst do not make it to the next generation. After selecting two parent chromosomes, a random crossover point is selected and two new chromosomes are produced by interchanging the ends of the parent ones. Then, mutation takes place, using a mutation probability for each gene in each chromosome. The iteration is completed after all new chromosomes are scored, based on their predictive performance. The loop terminates when the maximum number of generations is reached.

5.4 Experimental Design Algorithms

Factorial Design

A common experimental design is one with all input factors set at two levels each (high/low levels). A design with all possible high/low combinations of all the input factors is called a full factorial design in two levels. If there are k factors, each at 2 levels, a full factorial design has 2^k runs. Such an experiment allows studying the effect of each factor, as well as the effects of interactions between factors on the response variable. Factorial designs are commonly analysed using Analysis Of Variance (ANOVA) or regression analysis. When the number of factors increases, a fractional factorial design is preferred to reduce the time requirements.

Fractional Factorial Design

A factorial experiment in which only an adequately chosen fraction of the 'treatment' combinations (factors and levels combinations) required for the complete factorial experiment is selected to be run. Properly chosen fractional factorial designs have the desirable properties of being both balanced and orthogonal. Balanced designs are those with the same number of observations across experimental conditions, whereas orthogonal designs are those where at least the effects of one factor cancel out with the effects of another.

D-Optimal experimental design method

A D-optimal design is generated by an iterative search algorithm and seeks to minimize the covariance of the parameter estimates for a specified model. This is equivalent to maximizing the determinant *D* = $|X^TX|$, where *X* is the *Nxk* design matrix of model terms (*k* columns) evaluated at specific treatments in the design space (*N* rows).

A-Optimal experimental design method

The "exact" design problem is to find a *nxk* matrix Z, with rows selected from matrix *X (n* is a subset of *N)*, that is "best" in some sense. The algorithmic design calculation is with respect to *X*, and not to some larger space. The A-optimal design is generated by an iterative search algorithm which seeks to minimize □□□□□(□□)/□, where □□ $= (\frac{□'□}{□})^{-1}$ (Atkinson and Doven, 1992).

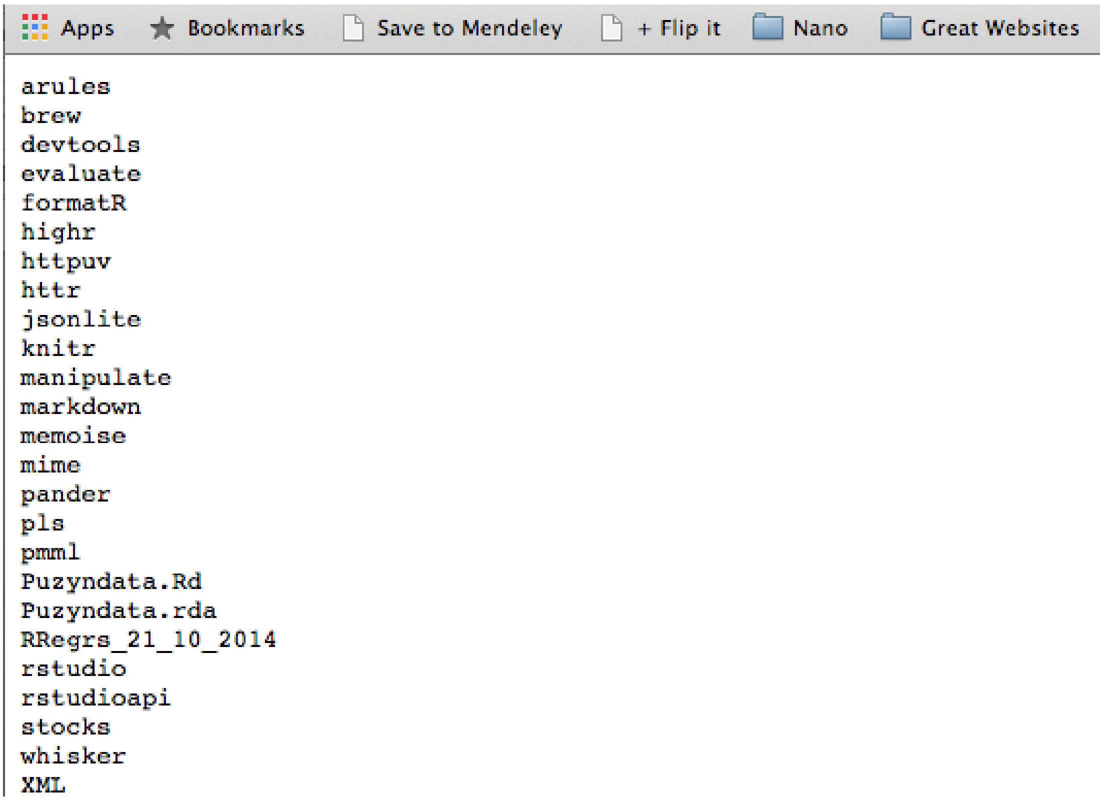
6. INTEGRATION OF R INTO THE ENANOMAPPER COMPUTATIONAL INFRASTRUCTURE- DEVELOPMENT OF A QSAR MODELLING PACKAGE IN THE R LANGUAGE

During the last decade, R (http://www.r-project.org/) has become the most popular language for computational statistics, visualization and data science, in both academia and industry (Smith 2014). Statisticians and data scientists use R to solve their most challenging problems in fields ranging from computational biology to quantitative marketing. Perhaps one of its most important advantages is that R users can easily find cutting-edge community-reviewed methods in statistics and predictive modelling from leading researchers in data science, free of charge.

Bioconductor, R's Bioinformatics branch, provides tools for the analysis and comprehension of high-throughput genomic data. Apart from providing free access to a broad range of powerful statistical and graphical methods for the analysis of genomic data, Bioconductor greatly facilitates the inclusion of biological metadata in the analysis of genomic data, e.g. literature data from PubMed (http://www.ncbi.nlm.nih.gov/pubmed), annotation data from Entrez genes, etc. This is one of its important features, since users can easily gather all the relevant biological information and analyse their integrated findings or validate their results.

Obviously, integration of R into the eNanoMapper system is of particular interest, but will require R to run on the server side. In order to achieve this task, we searched over alternatives and arrived to a technical solution that builds on OpenCPU (https://www.opencpu.org/) which defines an HTTP API for embedded scientific computing based on R although this approach should easily generalize to other computational back-ends (Ooms 2014). Particularly, OpenCPU is a JavaScript client library that integrates R with JavaScript. Two OpenCPU servers are available, namely the R package OpenCPU, and the OpenCPU cloud server. The first uses the httpuv web server to implement a single-user server that runs within an interactive R session on any platform, whereas the cloud server on the other hand is a multi-user implementation based on Ubuntu Linux and rApache (Ooms 2014, Ooms 2013). The latter yields much better performance and has advanced security and configuration options, but requires a dedicated Linux server. Another major difference between these implementations is how they handle concurrency. Because R is single threaded, httpuv handles only a single request at a time. Additional incoming requests are automatically queued and executed in succession using the same process. The cloud server on the other hand 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. Less flexible options are the StatET, an Eclipse based integrated development environment for R, and Renjin (http://www.renjin.org/) a java virtual machine interpreter for R. The former is dependent on Eclipse software, whereas the latter has limitations on the embedded R libraries.

OpenCPU provides users access to its services through RESTful web services. For example, the R packages the user has added are available through OpenCPU at the address http://147.102.82.122/ocpu/user/fidoli/library/, as shown in the screenshot of Figure 9.



```
arules
brew
devtools
evaluate
formatR
highr
httpuv
httr
jsonlite
knitr
manipulate
markdown
memoise
mime
pander
pls
pmml
Puzyndata.Rd
Puzyndata.rda
RRegrs_21_10_2014
rstudio
rstudioapi
stocks
whisker
XML
```

Figure 9 Listing of R packages available to the user in an example OpenCPU instance

As shown in Table 5, according to the OpenCPU API, (https://www.opencpu.org/api.html) the compatibility of OpenCPU to RESTful web services allows users to read objects or files, provide arguments for functions or run scripts. Please note that the paths should be appended to the server address, like http://147.102.82.122 should be appended to /ocpu/library/MASS/R/cats/json and the GET command should be sent to http://147.102.82.122/ocpu/library/MASS/R/cats/json .

Method	Target	Action	Arguments	Example
GET	object	read object	control output format	GET /ocpu/library/MASS/R/cats/json
POST	object	call function	function arguments	POST /ocpu/library/stats/R/rnorm
GET	file	read file	-	GET /ocpu/library/MASS/NEWS GET /ocpu/library/MASS/scripts/

POST	file	run script	control interpreter	POST /ocpu/library/MASS/scripts/ch01.R POST /ocpu/library/knitr/examples/knitr-minimal.Rmd

Table 5: Methods available through the OpenCPU API.

For example, the command to train a feed-forward neural network (FFNN) with a single hidden layer using the R package *nnet*, is the following:

```
curl http://147.102.82.122/ocpu/library/nnet/R/nnet
-d "x=[1,2,3,4,5]&y=[8,9,7,8,7]&size=2&abstol=0.00001"
    Input       output      nodes       calculations
    vector      vector      in hidden   tolerance
                            layer
```

This command would result in the following output, which provides the path where the results are available.

```
/ocpu/tmp/ x09bfba46db/R/.val
/ocpu/tmp/x09bfba46db/stdout
/ocpu/tmp/x09bfba46db/source
/ocpu/tmp/x09bfba46db/console
/ocpu/tmp/x09bfba46db/info
/ocpu/tmp/x09bfba46db/files/DESCRIPTION
```

These paths, when appended to the address of the R server provide information on the results. It should be noted here that the results are made available only for a predefined short period of time and then deleted. Therefore, they should be extracted by the user and stored elsewhere, as OpenCPU is not purposed for long-term storage of results. For example, a curl command on curl http://147.102.82.122/ocpu/tmp/x09bfba46db/console/text returns the information shown below. The same could be viewed by the user by accessing the address http://147.102.82.122/ocpu/tmp/x09bfba46db/console/text on a web browser.

```
> nnet(x = x, y = y, size = 2, abstol = 0.00001)
# weights:  7
initial  value 262.603678
final  value 234.000000
converged
a 1-2-1 network with 7 weights
options were -
```

Please note that /text was appended after the file path provided by OpenCPU (/ocpu/tmp/x09bfba46db/console) in order to get the results in text format. This information is also available in JSON format by appending /json:

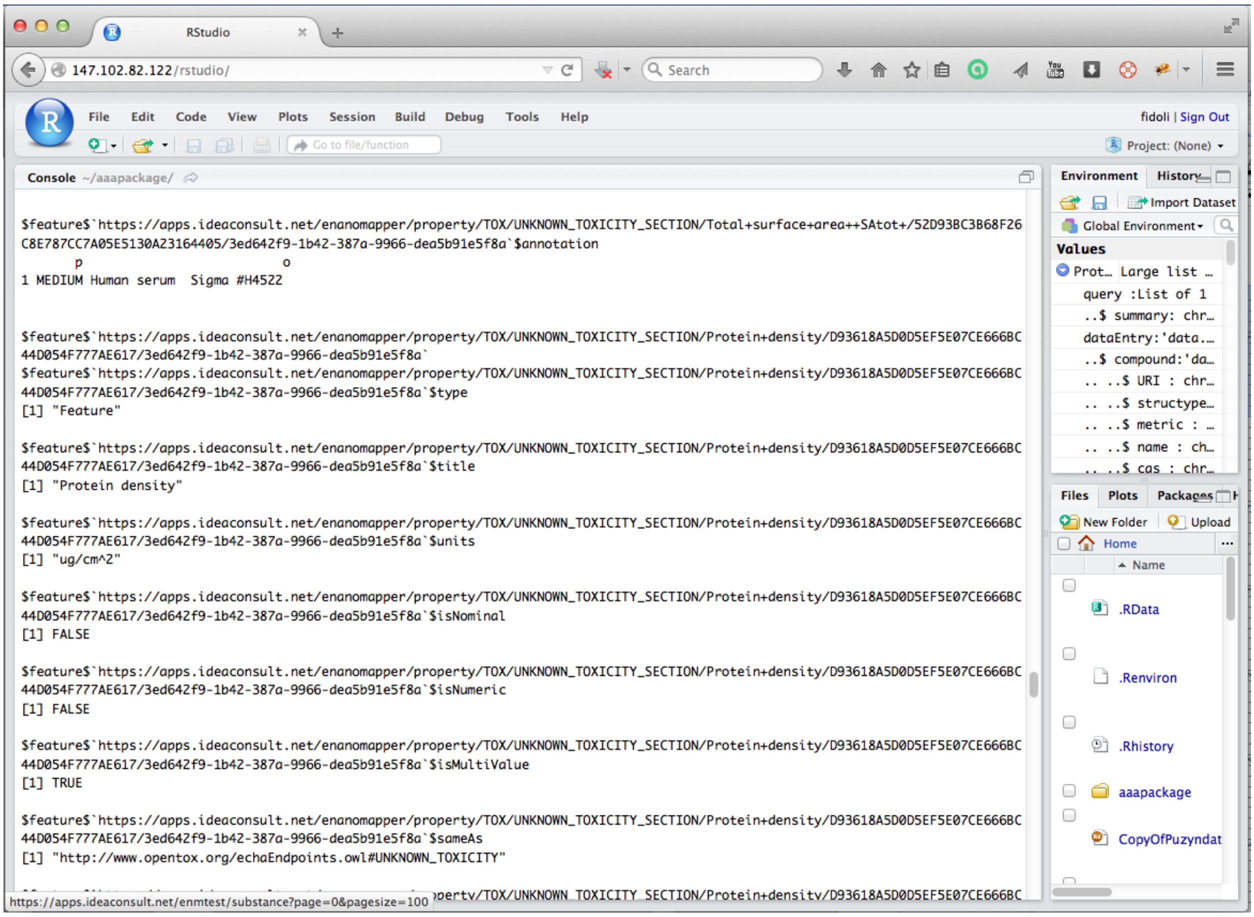
curl http://147.102.82.122/ocpu/tmp/x09bfba46db/console/json

```
[
  "> nnet(x = x, y = y, size = 2, abstol = 0.00001)",
  "# weights:  7\ninitial  value 262.603678 \nfinal  value 234.000000 \nconverged",
"a 1-2-1 network with 7 weights\noptions were -"
]
```

Except for the example where only a small dataset was given, users can input their data in various formats, such as JSON and csv files.

An alternative means of user interaction with R is the RStudio environment (RStudio 2012). Figure 10 shows the protein corona dataset from Walkey et. al. (2014) available within R, after being read from the eNanoMapper database at https://apps.ideaconsult.net/enmtest/substanceowner/FCSV-753FF8C6-DFAD-3E9F-8E89-D4500104AF33/dataset using the fromJSON command:

```
ProteinData <- fromJSON('https://apps.ideaconsult.net/enanomapper/substanceowner/FCSV-81ADF957-E54B-3BC8-A85A-53DC637572F6/dataset?media=application%2Fjson')
```



```
$feature$`https://apps.ideaconsult.net/enanomapper/property/TOX/UNKNOWN_TOXICITY_SECTION/Total+surface+area++SAtot+/52D93BC3B68F26
C8E787CC7A05E5130A23164405/3ed642f9-1b42-387a-9966-dea5b91e5f8a`$annotation
       p                          o
1 MEDIUM Human serum  Sigma #H4522


$feature$`https://apps.ideaconsult.net/enanomapper/property/TOX/UNKNOWN_TOXICITY_SECTION/Protein+density/D93618A5D0D5EF5E07CE666BC
44D054F777AE617/3ed642f9-1b42-387a-9966-dea5b91e5f8a`
$feature$`https://apps.ideaconsult.net/enanomapper/property/TOX/UNKNOWN_TOXICITY_SECTION/Protein+density/D93618A5D0D5EF5E07CE666BC
44D054F777AE617/3ed642f9-1b42-387a-9966-dea5b91e5f8a`$type
[1] "Feature"

$feature$`https://apps.ideaconsult.net/enanomapper/property/TOX/UNKNOWN_TOXICITY_SECTION/Protein+density/D93618A5D0D5EF5E07CE666BC
44D054F777AE617/3ed642f9-1b42-387a-9966-dea5b91e5f8a`$title
[1] "Protein density"

$feature$`https://apps.ideaconsult.net/enanomapper/property/TOX/UNKNOWN_TOXICITY_SECTION/Protein+density/D93618A5D0D5EF5E07CE666BC
44D054F777AE617/3ed642f9-1b42-387a-9966-dea5b91e5f8a`$units
[1] "ug/cm^2"

$feature$`https://apps.ideaconsult.net/enanomapper/property/TOX/UNKNOWN_TOXICITY_SECTION/Protein+density/D93618A5D0D5EF5E07CE666BC
44D054F777AE617/3ed642f9-1b42-387a-9966-dea5b91e5f8a`$isNominal
[1] FALSE

$feature$`https://apps.ideaconsult.net/enanomapper/property/TOX/UNKNOWN_TOXICITY_SECTION/Protein+density/D93618A5D0D5EF5E07CE666BC
44D054F777AE617/3ed642f9-1b42-387a-9966-dea5b91e5f8a`$isNumeric
[1] FALSE

$feature$`https://apps.ideaconsult.net/enanomapper/property/TOX/UNKNOWN_TOXICITY_SECTION/Protein+density/D93618A5D0D5EF5E07CE666BC
44D054F777AE617/3ed642f9-1b42-387a-9966-dea5b91e5f8a`$isMultiValue
[1] TRUE

$feature$`https://apps.ideaconsult.net/enanomapper/property/TOX/UNKNOWN_TOXICITY_SECTION/Protein+density/D93618A5D0D5EF5E07CE666BC
44D054F777AE617/3ed642f9-1b42-387a-9966-dea5b91e5f8a`$sameAs
[1] "http://www.opentox.org/echaEndpoints.owl#UNKNOWN_TOXICITY"

perty/TOX/UNKNOWN_TOXICITY_SECTION/Protein+density/D93618A5D0D5EF5E07CE666BC
```

Figure 10 The RStudio interface available in the browser window.

The OpenCPU functionality will be employed and integrated in the eNanoMapper infrastructure in order to make R's powerful capabilities, as well as any additional code (R scripts) or R packages created by eNanoMapper consortium members for tailor-made modelling solutions available through RESTful web services under the OpenTox API.

In particular, a new R package, called RRegrs will be developed that automates the creation of the best possible QSAR model (validated using certain criteria), by searching over many different algorithms and tuning the parameters in each algorithm. RRegrs will be largely dependent on the caret R package (http://topepo.github.io/caret/index.html), a library containing set of functions to streamline the predictive modelling process ranging from data scaling and normalization, to data partition and cross-validation techniques.

The full RRegrs code will include the following steps:

1. Loading dataset: data in csv format will be accepted,
2. Filter dataset: a regular check for highly correlated descriptors or descriptors with near zero variance will be conducted prior to scaling,
3. Scaling dataset: normalization and scaling options will be provided,
4. Feature selection: we will investigate the possibility to include embedded feature selection methods in the QSAR models or include methodologies (e.g. LASSO) which include a feature selection step,
5. Regression models: a selection of regression models will be included, for example, linear model (LM), Partial least squares regression (PLS), Support vector machines (SVM), offering the option of analysing the data using feature selection methods,
6. Summary with top models: all results from the selected QSAR models will be presented in output files,
7. Statistics of the best model: statistics of the best QSAR model, such as predictor values, coefficients, will be presented,
8. Y-randomization: the best QSAR model will be additionally tested using CV.

Whilst Implementation of RRegrs is in progress, its source code is currently hosted at GitLab (https://about.gitlab.com/). A detailed description of the RRegrs tool will be included in D4.3.

7. CONCLUSION

In this report we have described the technical specifications of the eNanoMapper analysis and modelling infrastructure. Necessary adjustments to OpenTox ontology and APIs are described, mainly exploiting the PMML capabilities. Additionally, descriptor calculation and developments are discussed by introducing Quantum Mechanical descriptors, Image derived descriptors, and Gene Ontology descriptors. Extended modelling methodologies as well as improvements and additions to modelling algorithms are introduced, covering predictive model building, model validation and, clustering methodologies. A technical solution for integrating the R computational language within the eNanoMapper infrastructure is described. This integration will allow the development of an automated tool for optimizing the creation of predictive nanoQSAR models. The infrastructure we have built and designed so far supports user's analysis and meta-analysis work, and for that reason may be subject to further development and enhancement during the lifecycle of the project and depending on partner's needs.

8. BIBLIOGRAPHY

1. Abraham, G.; Kowalczyk, A.; Loi, S.; Haviv, I.; Zobel, J. Prediction of breast cancer prognosis using gene set statistics provides signature stability and biological context. *BMC Bioinformatics* **2010**, *11*, 277 DOI: 10.1186/1471-2105-11-277.
2. Abràmoff, M. D.; Magalhães, P. J.; Ram, S. J. Image processing with imageJ. Biophotonics International, **2004**, 11, 36–41.
3. Aggarwal, C. C.; Gates, S. C.; Yu, P. S. On the merits of building categorization systems by supervised clustering. *Proc. fifth ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '99* **1999**, 352–356 DOI: 10.1145/312129.312279.
4. Atkinson, A.C.; Donev, A.N. *Optimum experimental designs*. Clarendon Press, 1992; p. 328.
5. Balbin, O. A.; Prensner, J. R.; Sahu, A.; Yocum, A.; Shankar, S.; Malik, R.; Fermin, D.; Dhanasekaran, S. M.; Chandler, B.; Thomas, D.; et al. Reconstructing targetable pathways in lung cancer by integrating diverse omics data. *Nat. Commun.* **2013**, *4*, 2617 DOI: 10.1038/ncomms3617.
6. Bansal, N.; Blum, A.; Chawla, S. Correlation Clustering. *Mach. Learn.* **2004**, *56*, 89–113 DOI: 10.1023/B:MACH.0000033116.57574.95.
7. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32 DOI: 10.1023/A:1010933404324.
8. Brown, I. D.; McMahon, B. CIF: The computer language of crystallography. *Acta Crystallogr. Sect. B Struct. Sci.* **2002**, *58*, 317–324 DOI: 10.1107/S0108768102003464.
9. Burello, E.; Worth, A. P. A theoretical framework for predicting the oxidative stress potential of oxide nanoparticles. *Nanotoxicology* **2011**, *5*, 228–235 DOI: 10.3109/17435390.2010.502980.
10. Burello, E.; Worth, A. P. A theoretical framework for predicting the oxidative stress potential of oxide nanoparticles. *Nanotoxicology*, **2011**, *5*, 228–235 DOI: 10.3109/17435390.2010.502980.
11. Cheng, Y.; Church, G. M. Biclustering of expression data. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **2000**, *8*, 93–103.
12. Chin, L.; Gray, J. W. Translating insights from the cancer genome into clinical practice. *Nature* **2008**, *452*, 553–563 DOI: 10.1038/nature06914.
13. Fedorov, V. V. *Theory of optimal experiments*; Elsevier Science, 1972; p. 306.
14. Fourches, D.; Pu, D.; Tassa, C.; Weissleder, R.; Shaw, S. Y.; Mumper, R. J.; Tropsha, A. Quantitative nanostructure-activity relationship modeling. *ACS Nano* **2010**, *4*, 5703–5712 DOI: 10.1021/nn1013484.
15. Fourches, D.; Pu, D.; Tassa, C.; Weissleder, R.; Shaw, S. Y.; Mumper, R. J.; Tropsha, A. Quantitative nanostructure-activity relationship modeling. *ACS Nano*, **2010**, *4*, 5703–5712 DOI: 10.1021/nn1013484.
16. Free Software Foundation, GNU General Public License https://www.gnu.org/copyleft/gpl.html (accessed Nov 1, 2014).
17. Gajewicz, A.; Rasulev, B.; Dinadayalane, T. C.; Urbaszek, P.; Puzyn, T.; Leszczynska, D.; Leszczynski, J. Advancing risk assessment of engineered nanomaterials: Application of computational approaches. *Advanced Drug Delivery Reviews*, 2012, *64*, 1663–1693.
18. Gajewicz, A.; Rasulev, B.; Dinadayalane, T. C.; Urbaszek, P.; Puzyn, T.; Leszczynska, D.; Leszczynski, J. Advancing risk assessment of engineered nanomaterials: Application of computational approaches. *Advanced Drug Delivery Reviews*, **2012**, *64*, 1663–1693.
19. Gajewicz, A.; Schaeublin, N.; Rasulev, B.; Hussain, S.; Leszczynska, D.; Puzyn, T.; Leszczynski, J. Towards understanding mechanisms governing cytotoxicity of metal oxides nanoparticles: Hints from nano-QSAR studies. *Nanotoxicology* **2014**, *5390*, 1–13 DOI: 10.3109/17435390.2014.930195.
20. Ge, C.; Du, J.; Zhao, L.; Wang, L.; Liu, Y.; Li, D.; Yang, Y.; Zhou, R.; Zhao, Y.; Chai, Z.; et al. Binding of blood proteins to carbon nanotubes reduces cytotoxicity. *Proceedings of the National Academy of Sciences*, **2011**, *108*, 16968–16973 DOI: 10.1073/pnas.1105270108

21. Hastie, T.; Tibshirani, R.; Friedman, J. *The elements of statistical learning data mining, inference, and prediction*; 2009; p. 745 S.
22. Indahl, U. G.; Liland, K. H.; Næs, T. Canonical partial least squares-a unified PLS approach to classification and regression problems. *J. Chemom.* **2009**, *23*, 495–504 DOI: 10.1002/cem.1243.
23. Kapralov, A. A.; Feng, W. H.; Amoscato, A. A.; Yanamala, N.; Balasubramanian, K.; Winnica, D. E.; Kisin, E. R.; Kotchey, G. P.; Gou, P.; Sparvero, L. J.; et al. Adsorption of surfactant lipids by single-walled carbon nanotubes in mouse lung upon pharyngeal aspiration. *ACS Nano* **2012**, *6*, 4147–4156 DOI: 10.1021/nn300626q.
24. Kapralov, A. A.; Feng, W. H.; Amoscato, A. A.; Yanamala, N.; Balasubramanian, K.; Winnica, D. E.; Kisin, E. R.; Kotchey, G. P.; Gou, P.; Sparvero, L. J.; et al. Adsorption of surfactant lipids by single-walled carbon nanotubes in mouse lung upon pharyngeal aspiration. *ACS Nano*, **2012**, *6*, 4147–4156 DOI: 10.1021/nn300626q.
25. Kaufman, L.; Rousseeuw, P. J. *Finding Groups in Data: an Introduction to Cluster Analysis*; John Wiley and Sons, **1990**; p. 342; DOI: 10.1002/9780470316801.
26. Kim, D.; Joung, J.-G.; Sohn, K.-A.; Shin, H.; Park, Y. R.; Ritchie, M. D.; Kim, J. H. Knowledge boosting: a graph-based integration approach with multi-omics data and genomic knowledge for cancer clinical outcome prediction. *J. Am. Med. Inform. Assoc.* **2014**, 1–10 DOI: 10.1136/amiajnl-2013-002481.
27. Lazzeroni, L., Owen, A. Plaid models for gene expression data http://statweb.stanford.edu/~owen/reports/plaid.pdf (accessed Nov 27, 2014).
28. Lesniak, A.; Fenaroli, F.; Monopoli, M. P.; Åberg, C.; Dawson, K. A.; Salvati, A. Effects of the presence or absence of a protein corona on silica nanoparticle uptake and impact on cells. *ACS Nano* **2012**, *6*, 5845–5857 DOI: 10.1021/nn300223w.
29. Malkiewicz, K.; Pettitt, M.; Dawson, K. A.; Hansson, S. O.; Lynch, I.; Lead, J. Nanomaterials in reach. *Toxicol. Lett.* **2011**, *205, Supplement*, S45 – DOI: http://dx.doi.org/10.1016/j.toxlet.2011.05.179.
30. Malkiewicz, K.; Pettitt, M.; Dawson, K. A.; Hansson, S. O.; Lynch, I.; Lead, J. Nanomaterials in reach. *Toxicol. Lett.*, **2011**, *205 Supplement*, S45 – DOI: http://dx.doi.org/10.1016/j.toxlet.2011.05.179.
31. Monopoli, M. P.; Walczyk, D.; Campbell, A.; Elia, G.; Lynch, I.; Baldelli Bombelli, F.; Dawson, K. A. Physical-Chemical aspects of protein corona: Relevance to in vitro and in vivo biological impacts of nanoparticles. *J. Am. Chem. Soc.* **2011**, *133*, 2525–2534 DOI: 10.1021/ja107583h.
32. Monopoli, M. P.; Walczyk, D.; Campbell, A.; Elia, G.; Lynch, I.; Baldelli Bombelli, F.; Dawson, K. A. Physical-Chemical aspects of protein corona: Relevance to in vitro and in vivo biological impacts of nanoparticles. *J. Am. Chem. Soc.,* **2011**, *133*, 2525–2534 DOI: 10.1021/ja107583h.
33. Ooms, J. The OpenCPU System : Towards a Universal Interface for Scientific Computing through Separation of Concerns. *arXiv* **2014**, 1–23.
34. Ooms, J. The RAppArmor Package : Enforcing Security Policies in R Using Dynamic Sandboxing on Linux. *J. Stat. Softw.* **2013**, *55*.
35. Pechter, R. What’s PMML and what's new in PMML 4.0? *ACM SIGKDD Explorations Newsletter*, **2009**, *11*, 19.
36. Puzyn, T.; Rasulev, B.; Gajewicz, A.; Hu, X.; Dasari, T. P.; Michalkova, A.; Hwang, H.-M.; Toropov, A.; Leszczynska, D.; Leszczynski, J. Using nano-QSAR to predict the cytotoxicity of metal oxide nanoparticles. *Nature nanotechnology*, **2011**, *6*, 175–178.
37. Resources, I. What is PMML ? Explore the power of predictive analytics and open standards. *IBM developerWorks*. **2010**, pp. 1–10 http://www.ibm.com/developerworks/library/ba-ind-PMML1/ba-ind-PMML1-pdf.pdf (accessed Oct 31, 2014).
38. Roduner, E. Size matters: why nanomaterials are different. *Chem. Soc. Rev.* **2006**, *35*, 583–592 DOI: 10.1039/b502142c.
39. Roduner, E. Size matters: why nanomaterials are different. *Chem. Soc. Rev.,* **2006**, *35*, 583–592 DOI: 10.1039/b502142c.
40. RStudio. RStudio: Integrated development environment for R. *The Journal of Wildlife Management*, **2012**, *75*.
41. Saber Hussain, Christin Grabinski, Nicole Schaeublin, Elizabeth Maurer, Mohan Sankaran, Ravindra Pandey, Jerzy Leszczynski, W. T. *Toxicity Evaluation of Engineered Nanomaterials: Risk Evaluation Tools (Phase 3 Studies)*; **2012**; pp. 1–55.
42. Sass, S.; Buettner, F.; Mueller, N. S.; Theis, F. J. A modular framework for gene set analysis integrating multilevel omics data. *Nucleic Acids Res.* **2013**, *41*, 9622–9633 DOI: 10.1093/nar/gkt752.

43. Smith, D. R is Hot Revolution Analytics. 2014 http://www.revolutionanalytics.com/whitepaper/r-hot (Aacessed Nov 27, 2014).
44. Subramanian, A.; Subramanian, A.; Tamayo, P.; Tamayo, P.; Mootha, V. K.; Mootha, V. K.; Mukherjee, S.; Mukherjee, S.; Ebert, B. L.; Ebert, B. L.; et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 15545–15550 DOI: 10.1073/pnas.0506580102.
45. Svetnik, V.; Liaw, A.; Tong, C.; Wang, T. Application of Breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. *Mult. Classif. Syst.* **2004**, 334–343 DOI: 10.1007/978-3-540-25966-4_33.
46. Tibshirani, R. Regression Selection and Shrinkage via the Lasso. *Journal of the Royal Statistical Society B*, 1996, *58*, 267–288.
47. Walkey, C. D.; Olsen, J. B.; Song, F.; Liu, R.; Guo, H.; Olsen, D. W. H.; Cohen, Y.; Emili, A.; Chan, W. C. W. Protein corona fingerprinting predicts the cellular interaction of gold and silver nanoparticles. *ACS Nano* **2014**, *8*, 2439–2455 DOI: 10.1021/nn406018q.
48. Winkler, D. a; Mombelli, E.; Pietroiusti, A.; Tran, L.; Worth, A.; Fadeel, B.; McCall, M. J. Applying quantitative structure-activity relationship approaches to nanotoxicology: current status and future potential. *Toxicology*, **2013**, *313*, 15–23 DOI: 10.1016/j.tox.2012.11.005.
49. Yang, X.; Regan, K.; Huang, Y.; Zhang, Q.; Li, J.; Seiwert, T. Y.; Cohen, E. E. W.; Xing, H. R.; Lussier, Y. A. Single sample expression-anchored mechanisms predict survival in head and neck cancer. *PLoS Comput. Biol.* **2012**, *8* DOI: 10.1371/journal.pcbi.1002350.
50. Zou, H.; Hastie, T. Regularization and variable selection via the Elastic Net. *J. R. Stat. Soc. Ser. B* **2005**, *67*, 301–320.

9. ANNEXES

	Name	Description	Platform	License	Website
1	**ABINIT**	ABINIT is a package whose main program allows calculations of total energy, charge density and electronic structure of systems made of electrons and nuclei within DFT, using pseudopotentials and a planewave or wavelet basis.	Windows Linux MAC OSX	GNU GPL	http://www.abinit.org/
2	**ACES**	ACES (Advanced Concepts in Electronic Structure) is a set of programs that performs Ab initio quantum chemistry calculations. The program is not intended for large scale HF-SCF or KS-DFT calculations.	Linux SGI	GNU GPL	http://www.qtp.ufl.edu/ACES/
3	**Adun**	Adun is a multipurpose, open source molecular simulation framework for biophysical and biochemical free-energy calculations	Linux MAC OSX	open source	http://adun.imim.es/
4	**Ascalaph**	Ascalaph is general-purpose molecular modelling software that performs quantum mechanics calculations for initial molecular model development, molecular mechanics and dynamics simulations in the gas or in condensed phase.	Windows Linux Mac OSX	GNU GPL	http://www.biomolecular-modeling.com/Ascalaph/index.html
5	**BigDFT**	A DFT massively parallel electronic structure code using a wavelet basis set. Wavelets form a real space basis set distributed on an adaptive mesh. Surfaces and isolated systems can be simulated with the proper boundary conditions.	Cross-platform	GNU GPL	http://bigdft.org
6	**COLUMBUS**	The Columbus Quantum Chemistry Programs is a collection of programs for high-level Ab initio molecular electronic structure calculations. The programs are designed primarily for extended multi-reference (MR) calculations on electronic ground and excited states of atoms and molecules.	Linux	Obtained free of charge	http://www.univie.ac.at/columbus
7	**CONQUEST**	A linear scaling, or O(N), DFT electronic structure code designed to perform DFT calculations on very large systems.	Linux	GNU GPL	http://www.order-n.org

8	**CP2K**	CP2K is Fortran 95 code to perform atomistic and molecular simulations of solid state, liquid, molecular, and biological systems. It provides a general framework for different methods such as DFT using a mixed GPW approach and classical pair and many-body potentials.	Fortran 95	GNU GPL	http://cp2k.org/
9	**DACAPO**	Dacapo is a total energy program based on density functional theory. The code may perform molecular dynamics / structural relaxation simultaneous with solving the Schrodinger equations within density functional theory.	Linux	GNU GPL	https://wiki.fysik.dtu.dk/dacapo
10	**DIRAC**	A program for Atomic and Molecular Direct Iterative Relativistic All-electron Calculations.	Windows Linux Mac OSX	Charge-free license agreements	http://www.diracprogram.org/doku.php
11	**Elk**	An all-electron full-potential linearized augmented-plane wave (FP-LAPW) code. Provides DFT calculations.	Linux	GNU GPL	http://elk.sourceforge.net/
12	**ErgoSCF**	A quantum chemistry program for large-scale self-consistent field calculations. Ergo (also called ErgoSCF) is an open source code for large scale HF and KS-DFT calculations, using linear scaling algorithms.	Linux	GNU GPL	http://www.ergoscf.org/
13	**ERKALE**	A quantum chemistry program used to solve the electronic structure of atoms, molecules and molecular clusters.	Windows Linux	GNU GPL	https://code.google.com/p/erkale
14	**Firefly**	Firefly (previously known as the PC GAMESS) is an Ab initio and DFT computational chemistry program.	Windows Linux Mac OSX	Charge free license agreements	http://classic.chem.msu.su/
15	**FreeON**	An experimental suite of programs for linear scaling quantum chemistry. FreeON performs HF, pure Density Functional, and hybrid HF/DFT calculations.	Linux Mac OSX	GNU GPL	http://www.freeon.org
16	**GAMESS**	The General Atomic and Molecular Electronic Structure System (GAMESS) is a program for Ab initio molecular quantum chemistry.	Windows Linux Mac OSX	Charge free license agreements	http://www.msg.ameslab.gov/gamess/index.html
17	**GPAW**	GPAW is a density-functional theory (DFT) Python code based on the projector-augmented wave (PAW) method and the atomic simulation environment (ASE).	Linux MAC OSX	-	https://wiki.fysik.dtu.dk/gpaw/

18	**JDFTx**	A plane-wave density functional code designed for Joint Density Functional Theory (JDFT), a framework for Ab initio calculations of electronic systems in contact with liquid environments.	Linux	GNU GPL	http://sourceforge.net/p/jdftx/wiki/Home/
19	**MADNESS**	Multiresolution Adaptive Numerical Environment for Scientific Simulation is a high-level environment for the solution of integral and differential equations in many dimensions.	Windows Linux Mac OSX	GNU GPL	https://github.com/m-a-d-n-e-s-s/madness
20	**MOPAC**	MOPAC (Molecular Orbital PACkage) is a semiempirical quantum chemistry program. MOPAC-based OpenTox web services already in place.	Windows Linux Mac OSX	Free and Commercial	http://cacheresearch.com/mopac
21	**MPQ**	The Massively Parallel Quantum Chemistry Program. It computes properties of atoms and molecules from first principles using the time independent Schrödinger equation.	Linux	GNU Lesser General Public License	http://www.mpqc.org
22	**Octopus**	Octopus Is a Scientific Program Aimed at the Ab Initio Virtual Experimentation. A pseudopotential real-space package aimed at the simulation of the electron-ion dynamics of one-, two-, and three-dimensional finite systems subject to time-dependent electromagnetic fields. The program is based on time-dependent DFT (TDDFT) in the Kohn-Sham scheme.	Linux	GNU GPL	http://www.tddft.org/programs/octopus
23	**OpenMX**	OpenMX (Open source package for Material eXplorer) is a software package for nano-scale material simulations based on density functional theories (DFT), norm-conserving pseudopotentials, and pseudo-atomic localized basis functions.	Linux	GNU GPL	http://www.openmx-square.org
24	**PSI**	An open-source suite of Ab initio quantum chemistry programs designed for efficient, high-accuracy simulations of a variety of molecular properties.	Windows Linux Mac OSX	GNU GPL	http://www.psicode.org

25	**Quantum ESPRESSO**	Quantum Espresso is an integrated suite of Open-Source computer codes for electronic-structure calculations and materials modelling at the nanoscale. It is based on density-functional theory, plane waves, and pseudopotentials.	Linux	Open source distribution	http://www.quantum-espresso.org/
26	**Wannier90**	A program for calculating maximally-localised Wannier functions (MLWF) from a set of Bloch energy bands that may or may not be attached to or mixed with other bands.	Windows Linux Mac OSX	GNU GPL	http://www.wannier.org
27	**Yambo Code**	Yambo Is a FORTRAN/C Code for Many-Body calculations in solid state and molecular physics.	Linux	GNU GPL	http://www.yambo-code.org/

Table 6: Software for Molecular Dynamic Simulations and DFT/Ab initio/Monte Carlo/Semi-empirical calculations

	Name	Description	Platform	License	Website
1	**Appion**	A comprehensive web interface and python scripting system for single-particle analysis, which allows performing the entire 3D-Electron Microscopy image processing work-flow, from micrograph preprocessing to 3D model refinement.	Linux	Open Source, Apache License 2.0	http://appion.org/ (web page was down on 19/12/14)
2	**Bio-Formats**	Bio-Formats is a Java library for reading and writing biological image files. The primary goal of Bio-Formats is to facilitate the exchange of microscopy data between different software packages and organizations.	Java Matlab toolbox ImageJ plugin	GNU GPL or commercial license from Glencoe Software	http://www.openmicroscopy.org/site/products/bio-formats
3	**Bsoft**	Bsoft (Bernard's Software Package) is a collection of programs and a platform for development of software for image and molecular processing in structural biology.	Mac OSX Unix	Open Source	http://bsoft.ws/

4	**CellProfiler**	Open-source cellular image analysis software. Can easily handle large batches of images and produce numerical data.	Mac OSX Unix Windows	GPL	http://cellprofiler.org/
5	**EMAN2**	A broadly based greyscale scientific image processing suite with a primary focus on processing data from transmission electron microscopes. Originally for single particle reconstructions at the highest possible resolution, the suite offers support for single particle Cryo-electron tomography, and tools useful in many other subdisciplines such as helical reconstruction, 2-D crystallography and whole-cell tomography. EMAN2 is capable of processing very large data sets (>100,000 particle) very efficiently.	Mac OSX Unix Windows	Open Source, GPL/BSD	http://blake.bcm.edu/emanwiki/EMAN2
6	Endrov	A multi-purpose image analysis program. Graphical scripting language, as well as traditional scripting. Access almost all commercial and open file formats.	All Platforms (Java) Plug-ins Can interact with Matlab.	BSD3 license	http://www.endrov.net
7	**Eos**	An extensible and general image analysis system for electron microscopy. It supplies numerous small tools for image analysis (including general image processing such as smoothing, labeling, binarization and EM-specific tools such as CTF correction, alignment, classification, 3D-reconstruction, map/PDB structural analysis and pseudo-atomic modeling), Integrated tools (for single particle analysis, helical reconstruction, electron tomography), Object-oriented libraries by C and prototype-source codes for tool developers.	Mac OSX Unix Windows	Creative Commons Attribution	http://www.yasunaga-lab.bio.kyutech.ac.jp/Eos
8	**Fiji**	Fiji is an image processing package that can be described as a distribution of ImageJ (and ImageJ2) together with Java, Java3D and a lot of plugins organized into a coherent menu structure.	All Platforms (Java) Mac OSX Unix Windows web or from 3rd party software	GNU GPL	http://fiji.sc/Fiji

9	**ImageJ**	ImageJ can calculate area and pixel value statistics of user-defined selections and intensity thresholded objects, measure distances and angles, create density histograms and line profile plots. Supports standard image processing functions, geometric transformations such as scaling, rotation and flips. The program supports any number of images simultaneously that the available memory can accommodate. Multithreaded architecture allows use of multi-CPU hardware. Supports many image formats and image stacks.	All Platforms (Java) Mac OSX Unix Windows web or from 3rd party software	Freely available and in the public domain. No license is required.	http://imagej.nih.gov/ij/
10	**IMAGIC**	A high end environment for the analysis of images, spectra and other multi-dimensional data-sets. IMAGIC's software package is aimed at processing (huge) data sets from (cryo-) electron microscopy, especially in the field of single particle analyses in Structural Biology.	Mac OSX Unix Windows	Freeware	https://imagescience.de/imagic.html
11	**IPLT**	The IPLT (Image Processing Library & Toolbox) is primarily designated for electron microscopy, with particular emphasis on 2D electron crystallography. It consists of several modular class libraries.	Mac OSX Unix Windows	Free/Open Source, GPL	http://www.iplt.org/
12	JMagick	An open source Java interface of ImageMagick. Implemented in the form of a thin Java Native Interface (JNI) layer into the ImageMagick API.	All Platforms (Java)	GNU Library or LGPLv2	https://github.com/techblue/jmagick
13	**Micrograph Data Processing Program (MDPP)**	The Biozentrum Micrograph Data Processing Program (MDPP) is a general purpose image processing package. MDPP is an image processing package designed for micrograph data. It contains program to process data in a number of ways including many averaging methods, statistical methods and reconstruction schemes.	Mac OSX Unix	Free, GPLv3	http://sourceforge.net/projects/mdpp/
14	**OMERO**	View, organize, analyze and share your data from anywhere you have internet	Mac OSX Unix Windows web		http://www.openmicroscopy.or

		access. Support for native image file formats, metadata. Contains ImageJ Plugin.	or from 3rd party software		g/site/products/omero
15	**OpenCV**	Its library includes several hundreds of computer vision algorithms, such as imgproc (an image processing module that includes linear and non-linear image filtering, geometrical image transformations, color space conversion, histograms) and features2d (salient feature detectors, descriptors, and descriptor matchers).	Mac OSX Unix Windows Android iOS	BSD	http://opencv.org/
16	**SIMPLE**	SIMPLE (Single-particle IMage Processing Linux Engine) 2.0 implements an ab initio reconstruction algorithm tailored to flexible, asymmetrical single-particles. Provides image clustering, ab initio 3D alignment, heterogeneity analysis, reconstruction, and refinement algorithms.	Mac OSX Unix	GNU GPL	http://simple.stanford.edu/
17	**Slicer**	A software platform for the analysis (including registration and interactive segmentation) and visualization (including volume rendering) of medical images and for research in image guided therapy.	Mac OSX Unix Windows	Open Source (BSD-style license)	http://www.slicer.org/
18	**SPARX**	SPARX (Single Particle Analysis for Resolution eXtension) is a new image processing environment with a particular emphasis on TEM structure determination. It includes a user interface that provides a graphical programming environment with a novel data/process-flow infrastructure, an library of python scripts that perform specific TEM-related computational tasks, and a core library of fundamental C++ image processing functions. SPARX relies on the EMAN2 library.	Mac OSX Unix Windows	joint BSD/GNU	http://sparx-em.org/sparxwiki
19	**SPIDER**	SPIDER (System for Processing Image Data from Electron microscopy and Related fields) is an image processing system for electron microscopy. Contains numerous operations for: 3D reconstruction, averaging of single particle macromolecule specimens, multivariate statistical classification of	Mac OSX Unix	Most of the source code in SPIDER is available under the GPL License	http://www.wadsworth.org/spider_doc/spider/docs/spider.html

		images, and electron tomography.			
20	**Suprim**	A flexible, modular software package intended for the processing of electron microscopy images. The system consists of a set of image processing tools or filters, written in the C programming language, and a command line style user interface based on the UNIX shell. The pipe and filter structure of UNIX and the availability of command files in the form of shell scripts eases the construction of complex image processing procedures from the simpler tools.	Unix	Open Source	ami.scripps.edu/software/suprim/ (web page was down on 19/12/14)
21	**Xmipp**	Xmipp, "X-Window-based Microscopy Image Processing Package", is a specialized suite of image processing programs, primarily aimed at obtaining the 3D reconstruction of biological specimens from large sets of projection images acquired by TEM. It is a comprehensive package for single-particle analysis, which allows performing the entire 3D-Electro Microscopy image processing work-flow, from micrograph preprocessing to 3D model refinement.	Mac OSX Unix Source code Virtual Machine	GPLv2	http://xmipp.cnb.csic.es/

Table 7: Image processing software relevant to nanoparticles