nano-lazar: Read across predictions for nanoparticle toxicities with calculated and measured properties

Christoph Helma[1], Micha Rautenberg[1], Denis Gebele[1]

December 16, 2016

Abstract

The lazar framework for read across predictions was expanded for the prediction of naoparticles, and a new methodology for calculating nanoparticle descriptors from core and coating structures was implemented. In order to compare nanparticle descriptor sets and local regression algorithms 60 independent crossvalidation experiments were performed for the Protein Corona dataset obtained from the eNanoMapper database. The best RMSE and r^2 results were obtained with protein corona descriptors and the weighted random forest algorithm, but its 95% prediction interval is significantly less accurate than models with simpler descriptor sets (measured and calculated nanoparticle properties). The most accurate prediction intervals were obtained with measured nanoparticle properties with RMSE and r^2 valus that show no statistical significant difference ($p < 0.05$) to the protein corona descriptors. Calculated descriptors are interesting for cheap and fast high-throughput screening purposes, random forest models have significantly lower r^2 values, but RMSE and prediction intervals are comparable to protein corona and nanoparticle random forest models.

[1] in silico toxicology gmbh, Basel, Switzerland

Introduction

Read across is a commonly used approach for the risk assessment of chemicals. Read across procedures are based on the assumption that similar compounds cause similar biological effects. In order to estimate the activity of a novel compound a researcher will search for similar compounds with known biological activities and deduce the activity of the new compound from this data. In order to make the procedure reproducible, traceable and objective the authors of this paper have developed a computer program (`lazar`, (Maunz et al. 2013)) that

automates the risk assessment process. The objective of the current study was to extend `lazar` for the risk assessment of nanomaterials.

The concept of chemical *similarity* is the key idea behind all read across procedures. But similarity is not an intrinsic property of substances, it can be defined in different ways and the utility and performance of similarity measures depends on each specific use case.

Structural similarity is most frequently used in the risk assessment of compounds with a well defined chemical structure. These similarity definitions are obviously not directly applicable to nanomaterials, because they lack a well defined structure. It is however relatively straightforward to adapt other concepts, e.g. similarity in terms of chemical properties or in terms of biological effects. Compared to structural similarity, which can be calculated directly from chemical structures, these similarity definitions depend on actual *measurements*, which makes their estimation more expensive and time consuming. For this reason we have developed a novel concept of structural similarity for nanomaterials, which is based on the chemical fingerprints of core and coating materials. According to our knowledge, this is the first time that nanoparticle toxicities have been predicted from calculated properties alone.

In order to estimate the utility of these similarity concepts for nanomaterials, we have performed model building and validation experiments for models based on

- structural similarity (based on core and coating fingerprints)
- property similarity (based on measured nanoparticle properties)
- biological similarity (based on the interaction with serum proteins)

and compared the local regression algorithms

- weighted average
- partial least squares
- random forests

In addition we intend to address the important topic of *reproducible research* with this publication. It is in our experience frequently impossible to reproduce computational experiments for a variety of reasons, e.g.

- publications lack important details about algorithms and data
- authors use proprietary software that does not disclose its algorithms
- original software, libraries and operating systems are not available anymore

Our attempt to address these problems is to provide a self contained environment that contains all software and data for the experiments presented in this manuscript. It contains also a build system for the manuscript, that pulls results and figures directly from validation experiments (similar to the `R knitr` package (Xie 2015)).

A self-contained system with the compiled manuscript and all libraries and external programs required for repeating the validation experiments is publicly available as a `docker` image from DockerHub (https://hub.docker.com/r/

insilicotox/nano-lazar-paper). Apart from repeating the experiments for this paper this image can also be used for extending the system, testing other descriptor and modelling algorithms and comparing validation results with the current benchmark.

Source code for the manuscript and validation experiments has been published under a GPL3 license at Github (https://github.com/opentox/nano-lazar-paper). The `lazar` framework library has been published under the same license (https://github.com/opentox/lazar).

A graphical webinterface for `nano-lazar` model predictions and validation results is publicly accessible at https://nano-lazar.in-silico.ch, source code for the GUI can be obtained from https://github.com/enanomapper/nano-lazar.

Github and DockerHub repositories are tagged with `nano-lazar-paper` to identify the software version that corresponds to the published paper. As this project is under continuous development, it is likely that some of the algorithms will change in the future. In this case it is relatively straightforward to identify differences with the versioning system or to use the submitted version as benchmark for further developments.

Methods

The following sections give a high level overview about `nano-lazar` algorithms. Readers interested in unambiguous algorithm definitions can refer to the source code links in the text.

Datasets

Nanoparticle characterisations and toxicities were mirrored from the eNanoMapper database (Jeliazkova et al. 2015) via its REST API (https://github.com/opentox/lazar/blob/development/lib/import.rb#L9-L118). At present only the *Net cell association* endpoint of the *Protein corona* dataset, has a sufficient number of examples (121) to create and validate read-across models, all other public nanoparticle endpoints have less than 20 examples, which makes them unsuitable for local (Q)SAR modelling and crossvalidation experiments.

Algorithms

For this study we have adapted the modular `lazar` (*la*zy *s*tructure *a*ctivity *r*elationships) read across framework (Maunz et al. 2013) for nanoparticle model development and validation.

lazar was originally developed for small molecules with a defined chemical structure and uses chemical fingerprints for the identification of similar compounds (*neighbors*). Nanoparticles in contrast do not have clearly defined chemical structures, but they can be characterised by their composition (core and coatings), measured properties (e.g. size, shape, physicochemical properties) or the interaction with biological macromolecules. Within `nano-lazar` we use these properties for the identification of similar nanoparticles (*neighbors*) and as descriptors for local QSAR models.

nano-lazar makes read-across predictions with the following basic workflow: For a given nanoparticle lazar

- searches in a database for similar nanoparticles (*neighbors*) with experimental toxicity data,
- builds a local QSAR model with these neighbors and
- uses this model to predict the activity of the query compound.

This procedure resembles an automated version of *read across* predictions in toxicology, in machine learning terms it would be classified as a *k-nearest-neighbor* algorithm (https://github.com/opentox/lazar/blob/development/lib/model.rb#L180-L257).

Apart from this basic workflow nano-lazar is completely modular and allows the researcher to use arbitrary algorithms for similarity searches and local QSAR modelling. Within this study we are using and comparing the following algorithms:

Nanoparticle descriptors

In order to find similar nanoparticles and to create local QSAR models it is necessary to characterize nanoparticles by descriptors. In this study we are using three types of descriptors:

Structural descriptors Calculated molecular fingerprints for core and coating compounds (MOLPRINT 2D fingerprints (Bender et al. 2004), *MP2D*, https://github.com/opentox/lazar/blob/development/lib/nanoparticle.rb#L17-L21)

Physico-chemical nanoparticle properties Measured nanoparticle properties from the eNanoMapper database (*P-CHEM*)

Biological nanoparticle properties Protein interaction data from the eNanoMapper database (*Proteomics*)

Nanoparticle fingerprints are a novel development for the characterisation of nanoparticles with well defined core and coating compounds. In this case it is possible to create molecular fingerprints for all of these compounds and use the union of these fingerprints as nanoparticle fingerprint. Based on our experience with small molecules we have selected MOLPRINT 2D fingerprints (Bender et al. 2004), which typically outperform predefined fingerprints (e.g. *MACCS*,

$FP4$) for (Q)SAR purposes. Despite its simplicity the concept works surprisingly well (see validation results) and enables toxicity predictions without measured properties. This can be useful e.g. for fast and cheap nanoparticle toxicity screening programs.

Feature selection

Calculated MP2D fingerprints are used without feature selection, as preliminary experiments have shown, that feature selection deteriorates the overall performance of read-across models (which is in agreement with our observations on small molecules).

Nanoparticle properties in the eNanoMapper database have not been measured for the purpose of read across and QSAR modelling. For this reason the database contains a lot of features that are irrelevant for toxicity. In preliminary experiments we have observed that using all available features for similarity calculations leads to neighbor sets that are unsuitable for local QSAR models, because large numbers of irrelevant features override the impact of features that are indeed relevant for toxicity.

For this reason we use the `lazar` concept of *activity specific similarities* (Maunz et al. 2013), by selecting only those features that correlate with a particular toxicity endpoint (Pearson correlation p-value < 0.05), which leads to a set of *relevant features*. This reduced feature set is used for similarity calculations and local QSAR models (https://github.com/opentox/lazar/blob/development/lib/feature_selection.rb#L6-L26). For crossvalidation experiments feature selection is repeated separately for each crossvalidation fold, to avoid overfitted models (Gütlein, Karwath, and Kramer 2012).

Neighbor identification

For binary features (MP2D fingerprints) we are using the union of core and coating fingerprints to calculate the Tanimoto/Jaccard index and a similarity threshold of $sim > 0.1$ (https://github.com/opentox/lazar/blob/development/lib/similarity.rb#L18-L20).

For quantitative features (P-CHEM, Proteomics) we use the reduced set of relevant features to calculate the *weighted cosine similarity* of their scaled and centered relevant feature vectors, where the contribution of each feature is weighted by its Pearson correlation coefficient with the toxicity endpoint. A similarity threshold of $sim > 0.5$ is used for the identification of neighbors for local QSAR models (https://github.com/opentox/lazar/blob/development/lib/similarity.rb#L37-L49).

In both cases nanoparticles that are identical to the query particle are eliminated from neighbors to obtain unbiased predictions in the presence of du-

plicates. (https://github.com/opentox/lazar/blob/development/lib/model.rb#L180-L257).

Local QSAR models and predictions

For read-across predictions local QSAR models for a query nanoparticle are build with similar nanoparticles (*neighbors*).

In this investigation we are comparing three local regression algorithms:

- weighted local average (WA, https://github.com/opentox/lazar/blob/development/lib/regression.rb#L6-L16)
- weighted partial least squares regression (PLS, https://github.com/opentox/lazar/blob/development/lib/caret.rb#L7-L78)
- weighted random forests (RF, https://github.com/opentox/lazar/blob/development/lib/caret.rb#L7-L78)

In all cases neighbor contributions are weighted by their similarity. The weighted local average algorithm serves as a simple and fast benchmark algorithm, whereas partial least squares and random forests are known to work well for a variety of QSAR problems. Partial least squares and random forest models use the `caret` R package (Kuhn 2008). Models are trained with the default `caret` settings, optimizing the number of PLS components or number of variables available for splitting at each RF tree node by bootstrap resampling.

Finally the local model is applied to predict the activity of the query nanoparticle. The RMSE of bootstrapped model predictions is used to construct 95% prediction intervals at 1.96*RMSE (https://github.com/opentox/lazar/blob/development/lib/caret.rb#L55-L77). Prediction intervals are not available for the weighted average algorithm, as it does not use internal validation.

If PLS/RF modelling or prediction fails, the program resorts to using the weighted average method.

Applicability domain

The applicability domain of `lazar` models is determined by the diversity of the training data. If no similar compounds are found in the training data (either because there are no similar nanoparticles or because similarities cannot be determined du to the lack of mesured properties) no predictions will be generated. Warnings are also issued, if local QSAR model building or model predictions fail and the program has to resort to the weighted average algorithm (https://github.com/opentox/lazar/blob/development/lib/model.rb#L180-L257).

The accuracy of local model predictions is indicated by the 95% prediction interval. (https://github.com/opentox/lazar/blob/development/lib/caret.rb#L55-L77).

Validation

For validation purposes we use results from 5 repeated 10-fold crossvalidations with independent training/test set splits (https://github.com/opentox/lazar/blob/development/lib/crossvalidation.rb#L85-L93). Feature selection is performed for each training dataset separately to avoid overfitting. For the same reason we do not use a fixed random seed for training/test set splits. This leads to slightly different results for each repeated crossvalidation run, but it allows to estimate the variability of validation results due to random training/test splits.

In order to identify significant differences between validation results, outcomes (RMSE, r^2, correct 95% prediction interval) are compared by ANOVA analysis, followed by Tukey multiple comparisons of means (https://github.com/enanomapper/nano-lazar-paper/blob/master/scripts/cv-statistics.rb).

Please note that recreating validations (e.g. in the Docker image) will not lead to exactly the same results, because crossvalidation folds are created randomly to avoid overfitting for fixed training/test set splits.

These five 10-fold crossvalidations are assigned to the final model, which is build from the complete training data. This validated model is used for further predictions, e.g. from the graphical webinterface.

Availability

Public webinterface: https://nano-lazar.in-silico.ch

Source code:

`lazar` framework: https://github.com/opentox/lazar

`nano-lazar` GUI: https://github.com/enanomapper/nano-lazar

Manuscript and validation experiments: https://github.com/opentox/nano-lazar-paper

Docker image with manuscript, validation experiments, `lazar` libraries and third party dependencies: https://hub.docker.com/r/insilicotox/nano-lazar-paper/

Results

The *Protein corona dataset* contains 121 Gold and Silver particles that are characterized by physchem properties (*P-CHEM*) and their interaction with proteins in human serum (*Proteomics*). In addition *MP2D* fingerprints were calculated for core and coating compounds with defined chemical structures.

Five repeated crossvalidations with independent training/test set splits were performed for the descriptor classes

- *MP2D* fingerprints (calculated, binary)
- *P-CHEM* properties (measured, quantitative)
- *Proteomics* data (measured, quantitative)
- *P-CHEM* and *Proteomics* data combined (measured, quantitative)

and the local regression algorithms

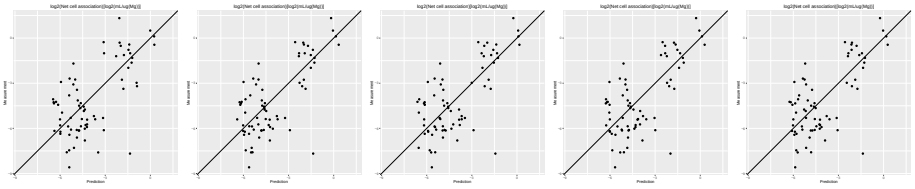
- local weighted average (*WA*)
- local weighted partial least squares regression (*PLS*)
- local weighted random forests (*RF*)

Results of these experiments are summarized in Table 1. Figure 1, Figure 2 and Figure 3 show the correlation of predictions with measurements for *MP2D*, *P-CHEM* and *Proteomics* random forests models. Correlation plots for all descriptors and algorithms are available as supplementary material (https://github.com/enanomapper/nano-lazar-paper/tree/master/figures). Table 2 lists *P-CHEM* properties of the Protein Corona dataset and their correlation with the *Net Cell Association* endpoint.

Table 1: Results from five independend crossvalidations for various descriptor/algorithm combinations. Best results (mean of 5 crossvalidations) are indicated by bold letters, statistically significant ($p < 0.05$) different results by italics. Results in normal fonts do not differ significantly from best results.

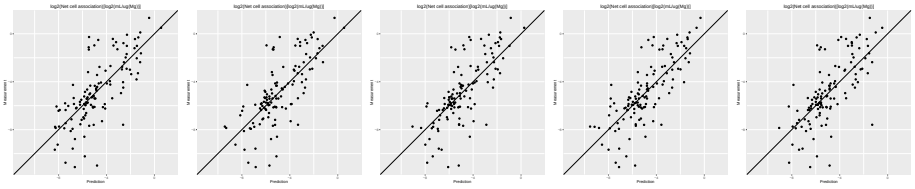
Descriptors	Algorithm	RMSE	r^2	% within prediction interval
MP2D	WA	*2.04 2.0 2.02 2.07 2.07*	*0.24 0.27 0.25 0.22 0.22*	NA
MP2D	PLS	*2.14 2.11 2.21 1.99 1.9*	*0.27 0.26 0.26 0.32 0.36*	94 97 91 91 97
MP2D	RF	1.84 1.67 1.68 1.69 1.71	*0.4 0.5 0.49 0.48 0.47*	94 96 96 94 94
P-CHEM	WA	*1.91 1.93 1.91 2.03 2.02*	*0.48 0.47 0.49 0.41 0.42*	NA
P-CHEM	PLS	*2.2 2.33 2.11 2.27 2.21*	*0.34 0.28 0.38 0.31 0.33*	97 92 96 93 91
P-CHEM	RF	1.8 1.82 1.77 1.68 1.86	0.54 0.53 0.56 0.6 0.51	**94 96 97 97 93**
Proteomics	WA	1.94 1.63 1.7 1.61 1.76	0.49 0.64 0.6 0.64 0.57	NA
Proteomics	PLS	1.67 1.63 1.86 1.74 1.8	0.62 0.64 0.53 0.59 0.56	*90 88 84 89 88*
Proteomics	RF	**1.66 1.69 1.81 1.68 1.6**	**0.62 0.61 0.57 0.6 0.65**	*89 89 89 87 89*
P-CHEM Proteomics	WA	1.61 1.56 1.71 1.66 2.41	0.64 0.66 0.6 0.62 0.33	NA

Descriptors	Algorithm	RMSE	r^2	% within prediction interval
P-CHEM	PLS	1.74 1.67 1.78	0.6 0.62 0.59	*91 90 86 85 86*
Proteomics		1.71 2.18	0.61 0.43	
P-CHEM	RF	*1.78 1.62 1.56*	*0.57 0.64 0.66*	*88 87 87 89 90*
Proteomics		*1.82 1.77*	*0.55 0.61*	



(a) (b) (c) (d) (e)

Figure 1: Correlation of predicted vs. measured values for five independent crossvalidations with *MP2D* fingerprint descriptors and local *random forest* models



(a) (b) (c) (d) (e)

Figure 2: Correlation of predicted vs. measured values for five independent crossvalidations with *P-CHEM* descriptors and local *random forest* models

Table 2: *P-CHEM* properties of the *Protein corona* dataset. Features correlating with the *Net cell association* endpoint (*relevant features*) are indicated by bold letters.

Property	Medium	Unit
Localized Surface Plasmon Resonance (LSPR) index	-	
Localized Surface Plasmon Resonance (LSPR) index	**Human serum**	
LSPR peak position (nm)	-	*nm*
Polydispersity index	**-**	*nm*

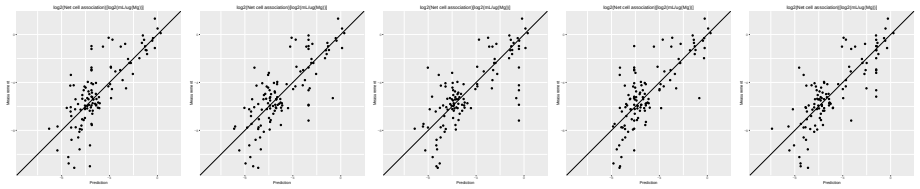
Property	Medium	Unit
Polydispersity index	Human serum	nm
Core size	**-**	nm
Autot (ICP-AES)	**Human serum**	$nmol$
Total surface area (SAtot)	Human serum	cm^2
Protein density	Human serum	ug/cm^2
Total protein (BCA assay)	**Human serum**	ug
ZETA POTENTIAL	**-**	mV
ZETA POTENTIAL	**Human serum**	mV
Z-Average Hydrodynamic Diameter	-	nm
Z-Average Hydrodynamic Diameter	**Human serum**	nm
Volume Mean Hydrodynamic Diameter	-	nm
Volume Mean Hydrodynamic Diameter	**Human serum**	nm
Number Mean Hydrodynamic Diameter	-	nm
Number Mean Hydrodynamic Diameter	Human serum	nm
Intensity Mean Hydrodynamic Diameter	-	nm
Intensity Mean Hydrodynamic Diameter	**Human serum**	nm

Discussion

Table 1 summarizes the results from five independent crossvalidations for all descriptor/algorithm combinations. The best results in terms of $RMSE$ and R^2 were obtained with *Proteomics* descriptors and local weighted *random forest* models. There are however six models without statistically significant differences in terms of $RMSE$ and five models in terms of r^2. The most accurate 95% prediction intervals were obtained with *P-CHEM* descriptors and *random forest* models, this models does not differ significantly from the best $RMSE$ and r^2 results.

Descriptors

In terms of descriptors the best overall results were obtained with *Proteomics* descriptors. This is in agreement with previous findings from other groups (Walkey et al. 2014, Liu et al. (2015), Papa et al. (2016), (**???**)). It is however interesting to note that the prediction intervals are significantly more inaccurate



(a) (b) (c) (d) (e)

Figure 3: Correlation of predicted vs. measured values for five independent crossvalidations with *Proteomics* descriptors and local *random forest* models

than those from other descriptors and the percentage of measurements within the prediction interval is usually lower than 90% instead of the expected 95%.

Using *P-CHEM* descriptors in addition to *Proteomics* does not lead to improved models, instead we observe an increased sensitivity towards training/test set splits (crossvalidation variability) and *random forest* results perform even significantly poorer than *Proteomics* descriptors alone.

P-CHEM descriptors alone perform surprisingly well, especially in combination with local *random forest* models, which does not show statistically significant differences to the best *Proteomics* model. On average more than 95% of the measurements fall within the 95% prediction interval, with significantly better results than for *Proteomics* descriptors. A summary of *P-CHEM* descriptors can be found in Table 2.

All *MP2D* models have poorer performance in terms of r^2, but the *random forest* model does not differ significantly in terms of $RMSE$ and measurements within the prediction interval.

Algorithms

With the exception of *P-CHEM/Proteomics* descriptors *random forests* models perform better than *partial least squares* and *weighted average* models with significant differences for *MP2D* and *P-CHEM* descriptors (detailed pairwise comparisons are available in the supplementary material). Interestingly the simple *weighted average* algorithm shows no significant difference to the best performing model for the *Proteomics* and *P-CHEM/Proteomics* descriptors.

Interpretation and practical applicability

Although *random forest* models with *Proteomics* descriptors havve the best performance in terms of $RMSE$ and r^2, the accuracy of the 95% prediction

interval is significantly lower than for *MP2D* and *P-CHEM* models (detailed pairwise comparisons in the supplementary material).

These problems seem to originate from the internal `caret` optimisation and validation algorithms which underestimate $RMSE$ values, that are used to calculate the prediction interval (see Algorithm section). And the observation that the *weighted average* algorithm, which does not use `caret`, performs comparatively well for *Proteomics* descriptors, supports this interpretation.

Our initial suspicion was that an unfavourable ratio between descriptors (785 before feature selection, 129 after feature selection) and training examples (121) causes this problem. $Randomforest$ and $partialleastsquares$ algorithms are on the other hand robust against a large number of descriptors and `caret` returns very realistic $RMSE$ values for *MP2D* fingerprints with a similar number of independent variables (100). For this reason it is at present still unclear, why the prediction intervals for *Proteomics* descriptors is significantly more inaccurate than for other descriptor types.

P-CHEM random forest models have the most accurate prediction interval and the $RMSE$ and r^2 performance is comparable to the $Proteomics$ model, although it utilizes a much lower number of descriptors (20 before feature selection, 10 after feature selection) which have not been measured for the purpose of (Q)SAR modelling. The main advantage from a practical point of view is that predictions of novel nanoparticles require a much lower amount of measurements than with $Proteomics$ data (although this argument may become obsolete with new high throughput techniques).

MP2D fingerprint descriptors are interesting from a practical point of view, because they do not require any measurements of nanoparticle properties. They need however defined chemical structures for core and coating compounds, which makes makes this approach infeasible for nanoparticle classes like carbon nanotubes. The resulting models do not differ significantly from the best results in terms of prediction accuracy ($RMSE$, measurements within prediction interval), but are significantly lower in terms of explained model variance (r^2). For practical purposes one may argue that the primary objective of read across models is to make accurate predictions and not to explain the model variance. For this reason we consider r^2 performance as secondary compared to $RMSE$ and prediction interval accuracies.

Currently a couple of (Q)SAR studies with global models have been published for the same dataset Walkey et al. (2014), Liu et al. (2015), Kamath et al. (2015), Papa et al. (2016), (**???**)], but unfortunately their results are not directly comparable, because we report results for the complete dataset with 121 gold and silver particles, while other authors report results for a subset of gold particles.

(Walkey et al. 2014) report leave-one-out (LOO) and 4-fold crossvalidation ($4CV$) results for 105 gold particles. They obtained the best results (LOO r^2 0.86, 4CV r^2 0.63) with partial least squares models, protein corona data with four additional physicochemical parameters and jackknife parameter selection.

Parameter selection was performed by crossvalidation, but it is unclear if parameters were selected on the complete dataset prior to LOO/4CV or separately for each LOO/4CV model. Performance wise the findings are roughly in agreement with our results. Assuming that feature selection was performed within crossvalidation folds we would expect 10-fold crossvalidation results between LOO and $4CV$ results. According to the authors the model developed for Gold compounds have little predictivity for Silver compounds, but a separate Silver model gave LOO r^2 of 0.79. $RMSE$ values are not available, although they are in our opinion more relevant for the predictive toxicology use case than r^2 values (prediction error vs explained model variance).

(Liu et al. 2015) report a 4CV r^2 of 0.843 for 84 Gold compounds using ϵ-support vector machines (ϵ-SVM) with 6 serum proteins and zeta potential as descriptors. Descriptors were selected with sequential forward floating selection. The methodological descriptions do not indicate explicitly, if feature selection was performed on the complete dataset or within 4CV folds. Judging from Figure 2 and the Methods section we have the strong impression that feature selection was performed prior to crossvalidation, which increases the likelihood of overfitted models, especially for aggressive feature selection schemes like *SFFS*. The 4CV $r2$ of 0.843 is clearly higher than our results, but it remains unclear, if the superior performance is due to better algorithms, a smaller more " regression friendly" dataset or overfitted models. Again we would have preferred $RMSE$ values for comparison purposes, which are unfortunately not available.

(Kamath et al. 2015) R2LOO=0.76 and R2LMO(25%)=0.72

(Papa et al. 2016) 84 gold compounds, r2 = 0.91; Q2loo = 0.81; r2ext = 0.79 selection of only six serum proteins Projection Pursuit Regression

Conclusion

We have performed 60 independent crossvalidation experiments for the Protein Corona dataset obtained from the eNanoMapper database in order to identify the best combination of descriptors for nanoparticle read across predictions. The best RMSE and r^2 results were obtained with protein corona descriptors and the weighted random forest algorithm, but its 95% prediction interval is significantly less accurate than models with simpler descriptor sets (measured and calculated nanoparticle properties). The most accurate prediction intervals were obtained with measured nanoparticle properties with RMSE and r^2 valus that show no statistical significant difference ($p < 0.05$) to the protein corona descriptors. Calculated descriptors are interesting for cheap and fast high-throughput screening purposes, they have significantly lower r^2 values than the best results, but RMSE and prediction intervals are comparable to the best results of our investigation.

For practical purposes we suggest to use nanoparticle properties when mea-

surements are available and the newly developed nanoparticle fingerprints for screening purposes without physicochemical measurements. Both models have been implemented with a graphical user interface which is publicly available at https://nano-lazar.in-silico.ch.

Acknowledgements

This work was performed as part of the EU FP7 project "Nanomaterials safety assessment: Ontology, database(s) for modelling and risk assessment Development of an integrated multi-scale modelling environment for nanomaterials and systems by design" (Theme NMP.2013.1.3-2 NMP.2013.1.4-1, Grant agreement no: 604134).

References

McDermott et al., 2013; Walkey et al., 2014 Yang et al., 2012; Balbin et al., 2013

Bender, Andreas, Hamse Y. Mussa, and Robert C. Glen, and Stephan Reiling. 2004. "Molecular Similarity Searching Using Atom Environments, Information-Based Feature Selection, and a Naïve Bayesian Classifier." *Journal of Chemical Information and Computer Sciences* 44 (1): 170–78. doi:10.1021/ci034207y.

Gütlein, Martin, Andreas Karwath, and Stefan Kramer. 2012. "CheS-Mapper - Chemical Space Mapping and Visualization in 3D." *Journal of Cheminformatics* 4 (1): 7. doi:10.1186/1758-2946-4-7.

Jeliazkova, Nina, Charalampos Chomenidis, Philip Doganis, Bengt Fadeel, Roland Grafström, Barry Hardy, Janna Hastings, et al. 2015. "The ENanoMapper Database for Nanomaterial Safety Information." *Beilstein J. Nanotechnol.*, no. 6: 1609–34. doi:doi:10.3762/bjnano.6.165.

Kamath, Padmaja, Alberto Fernandez, Francesc Giralt, and Robert Rallo. 2015. "Predicting Cell Association of Surface-Modified Nanoparticles Using Protein Corona Structure - Activity Relationships (Pcsar)." *Current Topics in Medicinal Chemistry* 15 (18): 1930–7. http://www.ingentaconnect.com/content/ben/ctmc/2015/00000015/00000018/art00015.

Kuhn, Max. 2008. "Building Predictive Models in R Using the Caret Package." *J. of Stat. Soft.*

Liu, Rong, Wen Jiang, Carl D. Walkey, Warren C. W. Chan, and Yoram Cohen. 2015. "Prediction of Nanoparticles-Cell Association Based on Corona Proteins and Physicochemical Properties." *Nanoscale* 7 (21). The Royal Society of Chemistry: 9664–75. doi:10.1039/C5NR01537E.

Maunz, Andreas, Martin Gütlein, Micha Rautenberg, David Vorgrimmler, De-

nis Gebele, and Christoph Helma. 2013. "Lazar: A Modular Predictive Toxicology Framework." *Frontiers in Pharmacology* 4. Frontiers Media SA. doi:10.3389/fphar.2013.00038.

Papa, E., J. P. Doucet, A. Sangion, and A. Doucet-Panaye. 2016. "Investigation of the Influence of Protein Corona Composition on Gold Nanoparticle Bioactivity Using Machine Learning Approaches." *SAR and QSAR in Environmental Research* 27 (7): 521–38. doi:10.1080/1062936X.2016.1197310.

Walkey, Carl D., Jonathan B. Olsen, Fayi Song, Rong Liu, Hongbo Guo, D. Wesley H. Olsen, Yoram Cohen, Andrew Emili, and Warren C. W. Chan. 2014. "Protein Corona Fingerprinting Predicts the Cellular Interaction of Gold and Silver Nanoparticles." *ACS Nano* 8 (3): 2439–55. doi:10.1021/nn406018q.

Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. http://yihui.name/knitr/.