Universal LD_{50} predictions using deep learning

Risa R. Sayre[1,2] & Christopher M. Grulke[1]

1. U.S. Environmental Protection Agency, Office of Research and Development, National Center for Computational Toxicology 2. ORISE Research Participant

Risa Sayre | ORCiD 0000-0002-6173-8020 | sayre.risa@epa.gov | 919-541-4871

EPA

www.epa.gov

Background and Purpose

EPA's National Center for Computational Toxicology (NCCT) develops and advances new alternative methods (NAMs) to support evaluation of the toxicity of thousands of chemicals to which Americans could be exposed. One such method is **QSAR** (quantitative structure-activity relationship); rigorous QSAR models can be used to predict the activity of new substances.

Purpose: Develop an approach using **deep learning** to create a **QSAR** to describe a set of oral rat **LD_{50} values** (a common metric used by risk assessors to determine the hazard posed by a chemical) provided as part of the Predictive Models for Acute Oral Systemic Toxicity project hosted by the ICCVAM Acute Toxicity Workgroup with to support hazard characterization for a broad range of chemicals.

We chose deep learning to build our QSAR for its ability to create robust models, even given the following difficulties :

- **Complexity**: many biological processes involved in the death of an animal
- **Outliers**: not all chemicals will elicit predictably similar responses
- **No assumption of domain knowledge**: prevention of analytical bias
- **Less sensitive to dimensionality of inputs**: flexibility of feature selection
- **Data uncertainty**: no metadata, so reproducibility cannot be assessed (Hagan 2014)

Methods

Here are the steps we took to generate our predictions:

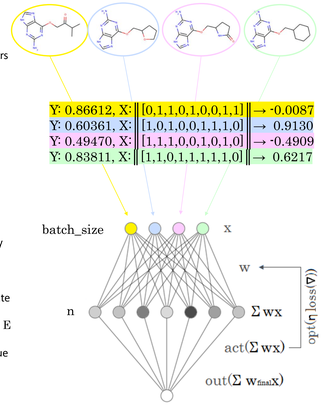
1. Calculate quantitative descriptors of the chemical structures
Using the cheminformatics tools Molecular Operating Environment (MOE), CORINA Symphony, and RDKit, we generated 0D (such as an atom count), 1D (fragment counts), and 2D (topostructural) descriptors from the provided canonical QSAR-ready SMILES

2. Clean and transform data (7660 rows) for optimal modeling
Y values (LD_{50} values)
- Log mol/kg LD_{50} values to make them more normally distributed
- Normalize from 0 to 1 for application to more functions

X values (structural descriptors)
- Discard rows with SMILES that did not generate mol
- For single-value descriptors, aggregate into vectors by source
- Scale vectors to zero mean with unit variance for function input

Divide into sets for training and validation of model

3. Develop neural network for each vectorized descriptor
- Take a random **batch_size** of chemical descriptors **x** and multiply them by random weights **w**
- Fit Σwx to the activation function **act** to find the error E in predicting the LD_{50} value
- Change **w** according to optimization function **opt** with learning rate **η** to minimize E as defined by loss function **loss**
- Keep updating the rates, reducing **η** when E stops dropping, until E < δ for 10 epochs
- Use final **w** in output function **out** to generate predicted LD_{50} value



↑ Simplified simulation of prediction value generation
Some examples of functions used ↓

4. Tune models
- Sweep all network hyperparameters to maximize R^2

  n: 71, 245, 2517 (based on various functions)
  batch_size: 10 through 1000
  act, **out**: linear, sigmoid, selu, relu, tanh, softmax, hard_sigmoid, softsign
  opt: Adadelta , Adagrad, Adam, Adamax, Nadam, RMSprop, SGD
  η: 0.0001, 0.001, 0.008, 0.01, 0.012, 0.1
  loss: binary_crossentropy, cosine_proximity, hinge, kullback_leibler_divergence, logcosh, mean_absolute_error, mean_absolute_percentage_error, mean_squared_error, mean_squared_logarithmic_error, squared_hinge, poisson

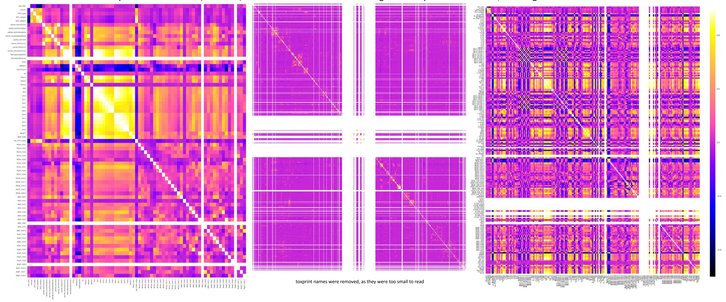
5. Identify most supported models for each descriptor
- Threshold modified from Golbraikh & Tropsha 2002 QSAR criteria

6. Create ensemble prediction from average of all models
- One model above threshold (if more than one, model with maximum $R^2_{validation}$) per descriptor

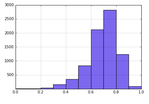
Metric	Description	Acceptable value
Training R^2	*variance in prediction given observation on training LD50s*	> 0.5
Validation R^2	*variance in prediction given observation on validation LD50s*	> 0.5
RMSE	*summed accuracy error over the LD50 set*	< $RMSE_{mean}$
Spearman ρ	*direction of association between descriptor and LD50*	> 0.6, p-val <0.01
γ_{rand} R^2	*prediction variance of the descriptors modeled in random LD50 values*	< 0.25

Dataset Evaluation

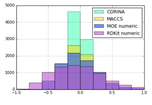
Distribution/variance of data
Linear correlation of descriptor values in **RDKit, CORINA, MOE**: the first four rows in each grid are dependent variables, showing near-zero linear correlation between X and Y



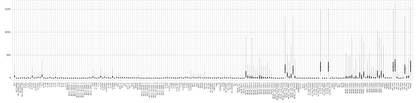
Normalized Y (log LD_{50}) values



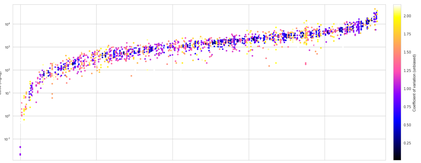
Selected scaled X values



Variability of individual values in Molecular Operating Environment descriptors



Measurement uncertainty in LD_{50} values
We calculated the precision and variance for the 560 substances for which there were 3 or more LD_{50} values. The mean coefficient of variation $\bar{c_v}^*$) of these values was 60%. The magnitude of the 95% confidence interval for many of the substances was greater than the span of values. The substances with a higher number of point estimates (as many as 48) did not have significantly more precise or less variable results. Without metadata, it is not possible to identify the source of uncertainty.
This measurement uncertainty bounds the performance of any LD_{50} model.



Uncertainty from parameterization with SMILES
All descriptors were calculated from SMILES (simplified molecular-input line-entry system), so several substances may have a common structure. The 41 multi-substances structures had a mean coefficient of variation of 62%. There are too few values to determine whether this is a significant source of additional uncertainty.

Skew
Y values: Skewed values are well modeled by some available algorithms.
X values: Although vectors are distributed normally once scaled, some highly variable values (shown at left) within those vectors are likely to contribute more heavily to descriptions, regardless of their importance in the system.

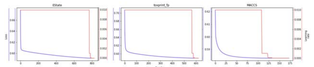
Results

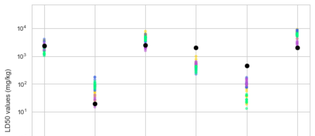
Hyperparameters and descriptors used in final models

descriptor	batch_size	act	opt	out	loss	Q2	R2
moe_numeric_param	42	softsign	Adagrad	sigmoid	binary_crossentropy	0.93	0.53
MACCS	43	softsign	Adagrad	sigmoid	binary_crossentropy	0.95	0.55
rdkit_numeric_param	60	softsign	Adadelta	hard_sigmoid	mean_absolute_error	0.78	0.50
toxprint_fp	46	softsign	SGD	hard_sigmoid	binary_crossentropy	0.86	0.50

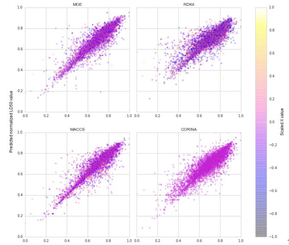
Algorithmic uncertainty
Stochastic optimization is variable by design. We used Monte Carlo dropout (Gal 2016) for cross-validation. The change in R^2 due to this type of uncertainty was normal for each model, with standard deviations between 0.004 and 0.012. The effect of algorithmic uncertainty (as well as the model error) on some point value estimates is shown at right; it is likely smaller than the measurement uncertainty.

Training performance
The model learned efficiently. The learning rate reduction and early stopping prevented overfitting, as intended.





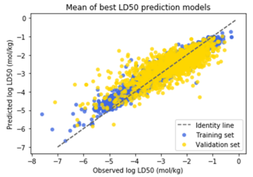
Effect of algorithmic uncertainty on point estimates ↑
← Relationships between error and learning rate



Effect of taking the mean for the consensus
The residual from the MACCS (Molecular Access System keys, calculated by RDKit) model was smaller than that from the mean 57.4% of the time. The R^2 of the consensus prediction were better than any single model (**0.644**).

Domain of applicability
X values from the center of each distribution (see *Selected scaled X values* plot above) did not lead to significantly better predictions on the set than those from the tails. We deemed all tested and predicted structures to be within the domain of applicability for this approach.

Degree of error in incorrect estimates
No model consistently over- or under-predicted. Only 2.6% of consensus predictions had a greater residual than comparing any value to the mean. The average normalized RMSE of included models was 0.069 (compare to the $RMSE_{mean}$ of 0.120).

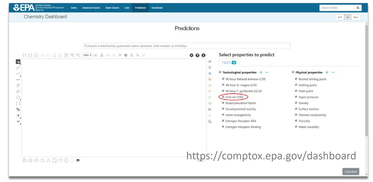


Conclusions & Discussion

We developed a deep learning QSAR model that predicted LD_{50} values above our target accuracy measures (R^2=0.644, RMSE=0.533)**.**

Availability, ease of use, degree of technical competency required to use
- Every tool used (except MOE) is openly available
- Written in a Jupyter notebook for portability
- Easy to add new descriptors for inclusion in the consensus by reading in a CSV file
- Requires very little user interaction and no expert knowledge to create predictions
- Previously trained models can be easily stored with HDF5 for future use. They do not take much space (<1000 KB)
- Creating new predictions for trained models takes only one line of code, and runs very quickly
- Mostly built on existing functions, so was fairly easy to develop
- Interpretation of results requires some statistical knowledge, but this is likely true with any method

Future work
Once fully optimized, this approach can be implemented on NCCT's Chemistry Dashboard. It would be easy to pipeline the model to run predictions for all substances with QSAR-ready SMILES to be stored as a chemical property. In the future, we hope it will be possible to quickly generate LD_{50} predictions just by drawing a structure.



Ideas for approach improvement
Data inclusion
- Add additional open source descriptors, like Padel
- Calculate 3D and other descriptors
- Identify inclusion criteria for LD_{50} to reduce noise
- Remove correlated descriptors, or descriptors that have the same value across the entire space to reduce training time
- Investigate correlation between descriptors and measurement variability

Data handling
- Use quantile normalization or scaling in log space to see if skew has an effect on calculation; normalize within vector
- Investigate if/how scaling and normalization reduce information on skewed data sets
- Does increasing vector size lead to R^2 inflation?
- Allow any combination of descriptors, regardless of source

Model building
- Investigate effect of tuning to hyperparameters other than R2
- Are there things that make a model hold more information than high description value?
- Include a bias term to allow the function to shift to fit the curves
- Initialize model weights to introduce the idea of asymmetry more quickly
- Create models for all individual descriptors or any combination of descriptors
- Create models for each GHS or EPA category, then develop point estimate from submodels
- Use built-in GridSearch function with randomized inputs rather than "priors"
- Use ensemble prediction (from some number of trials) for each descriptor type, allowing variation of hyperparameters
- Introduce weighting to each model's contribution to the consensus instead of an average

References

Abadi, M et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. tensorflow.org
Chollet, F et al. Keras; github.com/keras-team/keras
CORINA Symphony; 14698. Molecular Networks GmbH and Altamira LLC, 2017.
Gal Y. Uncertainty in Deep Learning. University of Cambridge, 2016.
Golbraikh A, Tropsha A. Beware of q2! *J Mol Graph Model.* 2002 Jan;20(4):269-76.
Hagan M, Demuth H, Beale M H, De Jesús O. Neural Network Design, 2nd Ed.
Heaton J. Introduction to Neural Networks for Java, 2nd Edition, 2008.
Molecular Operating Environment (MOE), 2016.08. Chemical Computing Group ULC, 2018.
Pedregosa *et al.* Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.
Python Software Foundation. Python Language Reference, version 3.5. www.python.org
RDKit: Open-source cheminformatics; www.rdkit.org

Thanks to John Wambaugh, Jeremy Fitzpatrick, Kate Saili, and R. Woodrow Setzer for providing feedback on this poster.

The approach and perspectives described in this poster do not necessarily reflect EPA policy.