Advice for Bad Computational Toxicologists

Srijit Seal*
Broad Institute of MIT and Harvard, Cambridge, Massachusetts, US, and University of Cambridge, Cambridge, Cambridgeshire, UK
ss2686@cam.ac.uk

Richard R. Rabbit (aka Thomas Hartung)*
Johns Hopkins University, Center for Alternatives to Animal Testing (CAAT), USA, and University of Konstanz, CAAT-Europe, Germany
thartun1@jhu.edu

Abstract

We recently gave advice for bad (traditional) toxicologists(Rabbit, 2025), —but we also have advice for the new breed of computational toxicologists. So many ways to do a lousy job…

Introduction

Well done! You are now enrolled in the art of building substandard toxicity-prediction models. Here, we promote cutting corners; genuine statistical diligence is purely optional. Follow these guaranteed shortcuts, and you'll never have to face real model validation: just grab any dataset, a cup of coffee, and apply the steps below!

1. Choose lots of duplicate compounds and split datasets randomly

Downloaded a new dataset? Why bother checking your dataset for duplicates hidden as tautomers or stereoisomers? Even better multiple entries for the same substance. Having the same compound appear in both your training and test sets is a fantastic way to improve your model's performance metrics.(Schaeffer, 2023) After all, training on the test data is all you need! Remember: the goal isn't to create models that generalize to new chemical space, but rather to memorize the training data perfectly. Reviewers love seeing an AUC-ROC value greater than 0.80!

2. Overinterpret PCA, UMAP, and t-SNE visualizations

Always present the chemical space as a principal component analysis (PCA) of Morgan fingerprints – do not bother if the variance explained is below 5%.(Walters, 2019) If you use t-SNE or UMAP visualizations(Irizarry, 2024; Social, n.d.), confidently declare meaningful clusters. Do you see some toxic compounds in one corner and some non-toxic compounds in another? This must indicate meaningful chemical distinctions – don't listen to those who say

these visualizations don't preserve global structure. Never mention that proximity in t-SNE or UMAP space doesn't necessarily translate to chemical similarity. The colorful plots look impressive regardless of their actual meaning. The old rule: If nobody believes you, add color!

3. Overfit models by using random splits

Scaffold- and time-based splits are such a hassle. Instead, simply create a single random train-test split and tune your hyperparameters using that same split repeatedly.(Walters, 2024) This approach ensures models that perform exceptionally well on your specific test.(Guo et al., 2024) Who needs generalizability when you can have perfect test set metrics? Try a few times and report the split that works. Some splits are real hits!

4. Ignore stratified splits entirely

Imbalanced datasets? No problem! Avoid stratified sampling at all costs—this ensures rare toxic classes are either absent from your training data or completely dominate your test set. Either way, your model will be unprepared for real-world chemical diversity. Bonus points if you never mention class distribution in your methods section!

5. Report metrics without confidence intervals

A proper bad computational toxicologist always reports AUC values with three decimal places but never with confidence intervals.(Seal, n.d.) Who cares if your test set contained only 20 compounds? That impressive 0.923 AUC must be meaningful! Never acknowledge that small test sets produce unstable metrics that could vary wildly with different splits. If anyone mentions a word, point them to the high F1 scores; after all, they wouldn’t know the F1 score for a model trained on a dataset with more toxic than nontoxic compounds, predicting all compounds as toxic is always greater than 0.67.

6. Celebrate high AUC

Brag about your model's impressive AUC, even when it correctly identifies only one class of toxic compounds. Always be grateful to the analogue bias. Conveniently ignore how the performance plateaus plummets thereafter. After all, the model correctly identifies 70 out of 100 toxic compounds, even though those 70 compounds are just analogues of each other! Who needs applicability domain analysis or metrics for comprehensive predictive power?

7. Use predicted probabilities as confidence scores

Always treat your model's output probabilities as direct measures of prediction confidence, even though this is not the case for most models, such as Random Forests and neural networks, which are notoriously poorly calibrated.(*Prediction Scores Are Not Probabilities*, n.d.) Tell your users that a prediction probability of 0.9 means "90% confidence"—they'll never know the difference until their critical decisions go spectacularly wrong.

8. Hide simple correlations behind complex feature names

Why admit your model is primarily leveraging molecular weight or logP when you can obfuscate these simple correlations behind impressive-sounding derived features? Terms like "POE_FSR" sound much more sophisticated than admitting your model mostly correlates toxicity with the number of neutral nitrogen atoms. Complex terminology distracts from the underlying triviality. Somehow, you have to justify the fancy computer your boss bought for you.

9. Skip baseline comparisons

Comparing your fancy deep learning model against simple baseline models is unnecessary busy work. Why check if a basic logistic regression model performs just as well? Your complex architecture with millions of parameters must be superior by default! Never risk discovering that your computational tour de force offers no meaningful improvement over simpler approaches, like predicting the mean activity from the training dataset.(Backenköhler et al., 2025) Average models are for average computational toxicologists!

10. Deploy models without retraining on the entire dataset

After spending months carefully constructing training, validation, and test splits, simply deploy the model you trained on just the training set. Why would you retrain on all available data before deployment? There is also no need to update models regularly—you're too busy chasing novelty. That would be efficient and scientifically sound! A proper bad computational toxicologist leaves much of the valuable data on the table and deploys suboptimal models.

Bonus. Cherry-picking Test Set is All You Need

The truly exceptional bad AI toxicologist knows the secret weapon: choose the test set based on a "preliminary analysis". Why leave evaluation to chance when you can engineer success? Use the entire dataset to identify features strongly correlated with your endpoint. Or better yet, train multiple models, identify compounds that consistently perform poorly, and simply remove these "problematic outliers" from your dataset. When reviewers ask about your methodology, vaguely mention "data cleaning", "curation" and "quality control." Remember, it's not cheating—it's creative curation!

Epilogue

This is a satirical piece, but one that may resonate with many who have been reviewer #2 (or suffered from reviewer #2). By highlighting these "bad" practices and adopting rigorous methodologies—from data curation to deployment—we aim to promote predictive models that genuinely support informed decision-making, reduce reliance on animal testing, and ultimately drive safer and more efficient toxicity prediction.

Real-world progress in toxicity prediction demands transparent data curation (deduplication, standardization, information and model leakage checks), robust validation (stratification, cross-vallidation, scaffold/time splits, confidence intervals), proper uncertainty quantification (calibration, full-curve analysis), meaningful benchmarks (simple baselines and ablations), and continual improvement (retraining on new data, model updates).(n.d.)

As scientists, we must commit to conducting reproducible, transparent, and trustworthy science for the benefit of communities.

References

Backenköhler, M., Groß, J., & Volkamer, A. (2025). Assay-based machine learning: Rethinking evaluation in drug discovery. In *ChemRxiv*. https://doi.org/10.26434/chemrxiv-2025-zsk6d

Guo, Q., Hernandez-Hernandez, S., & Ballester, P. J. (2024). *Scaffold Splits Overestimate Virtual Screening Performance*. arXiv. https://doi.org/10.48550/arXiv.2406.00873

Irizarry, R. (2024, December 23). *Biologists, stop putting UMAP plots in your papers*. Simply Statistics. https://simplystatistics.org/posts/2024-12-23-biologists-stop-including-umap-plots-in-your-papers/

Prediction Scores are not Probabilities. (n.d.). LearningKirvs. Retrieved April 24, 2025, from https://jtuckerk.github.io/prediction_probabilities.html

Rabbit, R. R., & Hartung, A. T. (2025). Advice for bad toxicologists. *NAM Journal*, *1*(100002), 100002.

Schaeffer, R. (2023). Pretraining on the test set is all you need. In *arXiv [cs.CL]*. arXiv. http://arxiv.org/abs/2309.08632

Seal, S. (n.d.). *pauc: Simple Python package to calculate ROC AUC with confidence intervals using DeLong's method*. Github. Retrieved April 24, 2025, from https://github.com/srijitseal/pauc

Social, @nikolay Oskolkov @oskolkov. (n.d.). *Is UMAP accurate? Addressing some fair and unfair criticism*. https://www.scilifelab.se/wp-content/uploads/2025/02/UMAP_NBIS_AI_IO_2025_Oskolkov.pdf

Walters, P. (2019, November 1). *Visualizing Chemical Space*.

https://practicalcheminformatics.blogspot.com/2019/11/visualizing-chemical-space.html

Walters, P. (2024, November 16). *Some Thoughts on Splitting Chemical Datasets*. https://practicalcheminformatics.blogspot.com/2024/11/some-thoughts-on-splitting-chemical.html

(n.d.). Seal, S. et al (2025). Machine Learning for Toxicity Prediction Using Chemical Structures: Pillars for Success in the Real World. https://doi.org/10.1021/acs.chemrestox.5c00033