Reproducibility in AI – a Case Study

Ting Li[1], Kamel Mansouri[2], Weida Tong[1]

[1]FDA/NCTR, Jefferson, AR, United States

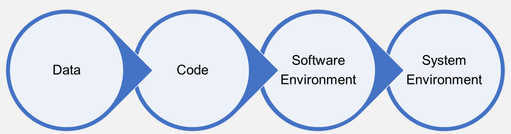
[2]NIH/NIEHS/DTT/NICEATM, Research Triangle Park, NC, United States

Introduction

- Reproducibility is essential in AI applications as it ensures consistent and reliable results. In regulatory contexts, reproducible AI models foster trust by allowing stakeholders to verify and validate outcomes. In this study, we used DeepCarc as a case study to evaluate factors influencing reproducibility.
- DeepCarc is a QSAR model designed to predict the carcinogenicity risk of chemical compounds, a critical factor in triggering regulatory actions for both new and existing substances.
- Traditional animal studies for carcinogenicity assessment are costly, time-consuming, labor-intensive, and raise ethical concerns. Additionally, it is impractical to conduct carcinogenicity tests on all compounds.
- In response, 21st-century toxicology has shifted towards alternative approaches, such as the 3Rs Principle (Replace, Reduce, Refine animal use) and the FDA's Predictive Toxicology Roadmap.
- While various QSAR models have been developed for carcinogenicity prediction, some are limited to specific chemical classes (e.g., aromatic amines, food-related phytochemicals), and others predict across broader classes but rely on carcinogenicity annotations from single species, such as rats. DeepCarc overcomes address these limitations.

Objectives

We **quantitatively** evaluated the reproducibility of DeepCarc through the following four components:



- **Data:** including dataset versions/sources, features, labels(output/target), training, validation and test set information.
- **Code:** including data preprocessing, model training, evaluation, algorithm hyperparameters, random seeds
- **Software environment:** the machine learning libraries (e.g., TensorFlow, PyTorch, Scikit-learn), python version, and package versions
- **System environment:** The hardware setup (e.g., GPU/CPU, memory) and operating system can impact the reproducibility of machine learning experiments, especially in cases where parallelization or random seed generation affects results.

Materials and methods

To develop the DeepCarc model, we utilized the NCTRlcdb, which consolidates multiple records—spanning gender, species, route of administration, and organ-specific toxicity—into a single carcinogenicity classification per compound, based on data from the Carcinogenic Potency Database. The DeepCarc model was then applied as a screening tool to assess carcinogenicity risk for 7,176 compounds from Tox21. Below is the study design for the DeepCarc model.

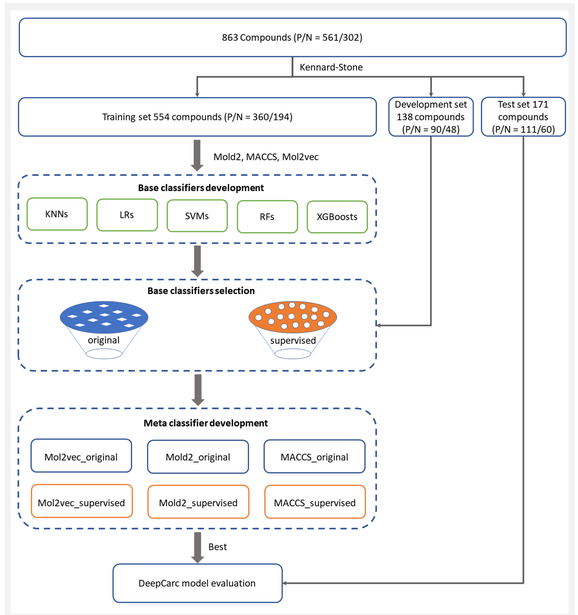
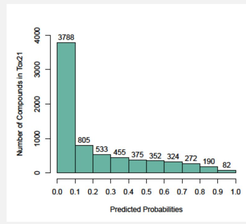


Figure 1. Overall workflow for the DeepCarc model including:
(1) Data preparation. 863 compounds were split into training (454 compounds), development (138 compounds), and test (171 compounds) sets based on the Kennard-stone algorithm.
(2) Base classifiers development. Five algorithms were used to develop the base classifiers from three different chemical representations, including Mol2vec, Mold2, and MACCS. Two base classifiers selection strategies were employed to select the optimized classifiers for meta classifier development.
(3) Meta classifier development. With three chemical representations and two selection methods, six groups of base classifiers, including Mol2vec_supervised, Mol2vec_original, Mold2_supervised, were used Mold2_original, MACCS_supervised, and MACCS_original. The probability prediction from selected base classifiers was used to train the neural network.
(4) Model evaluation. The DeepCarc model was evaluated on the test set.

Results and discussion

We evaluated the performance of the DeepCarc model using a test set consisting of 111 carcinogens and 60 non-carcinogens. The model achieved an accuracy of 0.754, an AUC of 0.776, and an MCC of 0.432. Additionally, DeepCarc was employed to screen the carcinogenicity potential of compounds from the Tox21 dataset, as shown in the following figure.



- Carcinogenicity of 7,176 compounds were evaluated.
- 1220 (17%) compounds were indicated with carcinogenic risk probability >= 0.5.
- 82 (1.14%) compounds were indicated with high carcinogenic risk probability >= 0.9.

Figure 2: The distribution of predicted carcinogenicity risk for Tox21 compounds.

The reproducibility of DeepCarc was assessed by evaluating its performance on the test set and predictions for the Tox21 compounds. We identified and listed the contributing factors affecting reproducibility for each component and presented the results along with potential solutions to mitigate the impact of these factors.

1. Without shared data, it is impossible to reproduce the DeepCarc results.

Factors	Reproducible?	Solutions
• Data availability • Data preprocessing • Data usage (training, test, validation)	• N/A	• Share data (GitHub (<2G), Zenodo(<50G), Dryad(>50G)) • Well document in data preprocessing • Record data usage (Avoid sharing data usage by sharing random seeds)

2. Without shared code, it is not able to reproduce the DeepCarc results.

Factors	Reproducible?	Solutions
• Code availability • Hyperparameters • Random seeds	• N/A	• Share code (e.g., GitHub) • Well document in Methods • Save classifiers with random seeds (e.g., .pkl file)

3. Software, like the package versions, could impact the reproducibility.

Factor	Reproducible?	Solution
• Packages version inconsistency (python, numpy, pandas, keras, scikit-learn, etc.)	• Performance metrics values matched on the test set • Tox21:1168/7176 match in probability prediction with 3 decimals (max difference: 0.110) • Tox21:7124 (52)/7176 match in binary classification	• Provide code along with the coding environment (e.g., Docker)

4. System environment, like the GPU, CPU, could slightly impact the reproducibility.

Factor	Reproducible?	Solutions
• Computing system (e.g., Linux, Windows)	• Performance metrics values matched on the test set • Tox21:2757/7176 match in probability prediction with 3 decimals (max difference: 0.043) • Tox21:7169 (7)/7176 match in binary classification	• Provide computing system parameters as specific as possible (System, GPU/CPU, etc.) • Can we accept this difference?

Conclusion

- Data and source code form the foundation of reproducibility in AI.
- Consistency in the software environment ensures results remain within an acceptable range of variation.
- System environment, while less impactful, also contributes to reproducibility.

To enhance reproducibility in AI methods, we recommend including both code and computational system parameters, and propose using a Docker strategy to ensure consistent AI-generated results.

Disclaimer

This poster reflects the views of the authors and does not necessarily reflect those of the Food and Drug Administration. Any mention of commercial products is for clarification and is not intended as an endorsement.