PyDrugLogics: A Python Package for Predicting Drug Synergies Using Boolean Models

Laura Szekeres ⬥[1] and **John Zobolas** ⬥[1]

1 Department of Cancer Genetics, Institute for Cancer Research, Oslo University Hospital, Oslo, Norway ROR

Summary

For complex diseases such as cancer, combined drug therapies can enhance the efficacy of personalized treatment and minimize side effects (Jin et al., 2023). Combined drug therapies may have a stronger effect than single-drug treatments, a concept referred to as synergy (Berenbaum, 1989). However, finding synergistic drug combinations is a challenging area of modern medicine. The vast number of possible drug combinations, even for small sets of drugs, makes it a complex problem due to the combinatorial growth in possibilities. This fact necessitates longer testing time in the laboratory experiments and a significant amount of experimental costs.

PyDrugLogics is a Python package that generates optimized Boolean models that represent a biological system and performs in-silico perturbations of these models to predict synergistic drug combinations. The implemented method is derived from the pipeline published by Flobak et al. (2023).

Statement of Need

Logical modeling is a powerful tool that can be used to reduce the costs of identifying synergistic drug combinations. By formalizing biological networks into logical models, we are able to simulate the behaviour of large-scale signaling networks and predict responses to perturbations (Eduati et al. (2020), Niederdorfer et al. (2020), Béal et al. (2021)).

Within this approach, the modeling process constructs a Boolean network to simulate the biological system, such as a cancer cell, and identify stable states that reflect the system's long-term behaviour. During the optimization process, the network's rules and topology are systematically adjusted to match the experimental steady states (e.g., protein activities). Multiple Boolean models are generated from this calibration process. This model ensemble is used to simulate the in-silico effects of drug perturbations and predict synergy scores for each combination. These predicted synergy scores are validated using experimentally measured outcomes, ensuring their predictive accuracy.

Several previous tools have addressed the challenges of modeling biological networks with logical modeling approaches. For instance, CellNOptR (Terfve et al., 2012) trains protein signaling networks to experimental data using multiple logic formalisms. In contrast, our tool focuses on optimizing Boolean models for predicting synergy scores. Another pipeline that was introduced in Dorier et al. (2016) uses Boolean models and a genetic algorithm for network construction and perturbation analysis, focusing on attractor identification and implemented as a command-line tool. PyDrugLogics builds on similar principles by leveraging advanced computational libraries such as MPBNs (Chatain et al., 2018) and PyBoolNet (Klarner et al., 2017) to calculate stable states and trap spaces, with the added benefit of a Python-based implementation.

The DrugLogics software pipeline was an important step forward for simulating biological networks with logical models and predicting synergy scores (Zobolas, 2020). Originally, structured as three separate Java packages, it preserved modular clarity. The Java implementation was well-designed and robust; nevertheless, the Java and Maven environment presented challenges in terms of maintainability, installation, and integrability with other community tools (Naldi et al., 2018).

Noticing these limitations, PyDrugLogics provides a practical solution that not only retains the core functionality of the Java-based pipelines, but also significantly boosts it by reducing the code complexity, improving the execution time, and introducing new features that expand its capabilities. By unifying the functionality of the three Java packages into a single Python package, PyDrugLogics simplifies the installation and software maintenance. Example improvements include the use of a standardized format, BoolNet (Müssel et al., 2010), for loading the models, as well as visualization options (e.g., precision-recall (PR) and receiver operating characteristic (ROC) curves) and statistical analyses (e.g., repeated subsampling of the ensemble Boolean models) for robust evaluation of prediction performance. Additional examples and comparisons of the Python and Java pipelines are available on the project wiki (Szekeres, 2025). PyDrugLogics provides an easy-to-use, flexible, and up-to-date solution for simulating Boolean networks and predicting synergistic drug combinations for the prioritization of follow-up lab experiments.

There has been a growing focus on developing tools that prioritize accessibility, reproducibility, and seamless integration in the logical modeling community. In particular, the CoLoMoTo Interactive Notebook aims to simplify integration and enables faster collaboration (Naldi et al., 2018). PyDrugLogics adopts this approach by integrating into the CoLoMoTo Docker, enabling compatibility with other tools so that researchers can combine methodologies and share results more effectively. The comprehensive documentation for PyDrugLogics is provided on the package website (Szekeres, 2024a), along with a detailed tutorial (Szekeres, 2024b). The package is available on PyPI (Szekeres, 2024c), offering a simple installation process and integration into Python workflows.

Brief Overview

The PyDrugLogics pipeline involves two main stages: calibration and prediction.

The calibration (`train`) function is responsible for loading a Boolean model for a particular biological system (e.g., a cancer cell) or the interactions for automatically constructing such a model. The next step is the optimization process that uses the PyGAD Genetic Algorithm (Gad, 2023) for finding the best set of Boolean models that fit the training data (e.g., protein measurements of the cancerous cell). The optimization process changes the model's operators and topology to ensure its behaviour fits the training data.

In the prediction (`predict`) function, the calibrated models are used to perform in-silico perturbations to simulate the effect of various drug treatments and their combinations. The perturbations represent changes to the system to mimic drug effects such as inhibition or activation of the affected proteins. The results of the perturbations are analyzed, and the predicted viability scores, which represent the system's response to drug treatments, are computed. Synergy scores are then derived from these viability scores, quantifying drug interactions and classifying combinations by synergistic potential. To verify the accuracy of the predictions, the pipeline requires the knowledge of the observed synergies, which serve as the ground truth, and they are typically derived from experimental datasets or literature sources. Using the observed synergies (binary labels: 0 for non-synergistic and 1 for synergistic) as the ground truth, binary classification metrics such as the ROC and PR AUC (area under the curve) are generated to evaluate how well the predicted synergy scores distinguish between synergistic and non-synergistic drug combinations.

Acknowledgements

References

Béal, J., Pantolini, L., Noël, V., Barillot, E., & Calzone, L. (2021). Personalized logical models to investigate cancer response to BRAF treatments in melanomas and colorectal cancers. *PLOS Computational Biology*, *17*(1), e1007900. https://doi.org/10.1371/journal.pcbi.1007900

Berenbaum, M. C. (1989). What is synergy? *Pharmacological Reviews*, *41*(2), 93–141. https://doi.org/10.1016/S0031-6997(25)00026-2

Chatain, T., Haar, S., Kolčák, J., & Paulevé, L. (2018). *Most permissive semantics of Boolean networks*. arXiv. https://doi.org/10.48550/ARXIV.1808.10240

Dorier, J., Crespo, I., Niknejad, A., Liechti, R., Ebeling, M., & Xenarios, I. (2016). Boolean regulatory network reconstruction using literature based knowledge with a genetic algorithm optimization method. *BMC Bioinformatics*, *17*(1), 410. https://doi.org/10.1186/s12859-016-1287-z

Eduati, F., Jaaks, P., Wappler, J., Cramer, T., Merten, C. A., Garnett, M. J., & Saez-Rodriguez, J. (2020). Patient-specific logic models of signaling pathways from screenings on cancer biopsies to prioritize personalized combination therapies. *Molecular Systems Biology*, *16*(2), e8664. https://doi.org/10.15252/msb.20188664

Flobak, Å., Zobolas, J., Vazquez, M., Steigedal, T. S., Thommesen, L., Grislingås, A., Niederdorfer, B., Folkesson, E., & Kuiper, M. (2023). Fine tuning a logical model of cancer cells to predict drug synergies: Combining manual curation and automated parameterization. *Frontiers in Systems Biology*, *3*, 1252961. https://doi.org/10.3389/fsysb.2023.1252961

Gad, A. F. (2023). PyGAD: An intuitive genetic algorithm python library. *Multimedia Tools and Applications*, *82*, 30895–30917. https://doi.org/10.1007/s11042-023-17167-y

Jin, H., Wang, L., & Bernards, R. (2023). Rational combinations of targeted cancer therapies: Background, advances and challenges. *Nature Reviews Drug Discovery*, *22*(3), 213–234. https://doi.org/10.1038/s41573-022-00615-z

Klarner, H., Streck, A., & Siebert, H. (2017). PyBoolNet: A python package for the generation, analysis and visualization of Boolean networks. *Bioinformatics*, *33*(5), 770–772. https://doi.org/10.1093/bioinformatics/btw682

Müssel, C., Hopfensitz, M., & Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, *26*(10), 1378–1380. https://doi.org/10.1093/bioinformatics/btq124

Naldi, A., Hernandez, C., Levy, N., Stoll, G., Monteiro, P. T., Chaouiya, C., Helikar, T., Zinovyev, A., Calzone, L., Cohen-Boulakia, S., Thieffry, D., & Paulevé, L. (2018). The CoLoMoTo interactive notebook: Accessible and reproducible computational analyses for qualitative biological networks. *Frontiers in Physiology*, *9*, 680. https://doi.org/10.3389/fphys.2018.00680

Niederdorfer, B., Touré, V., Vazquez, M., Thommesen, L., Kuiper, M., Lægreid, A., & Flobak, Å. (2020). Strategies to enhance logic modeling-based cell line-specific drug synergy prediction. *Frontiers in Physiology*, *11*, 862. https://doi.org/10.3389/fphys.2020.00862

Szekeres, L. (2024a). *PyDrugLogics package documentation*. GitHub Pages. https://druglogics.github.io/pydruglogics/

Szekeres, L. (2024b). *PyDrugLogics package Jupyter notebook tutorial*. GitHub. https://github.com/druglogics/pydruglogics/blob/main/tutorials/pydruglogics_tutorial.ipynb

Szekeres, L. (2024c). *PyDrugLogics PyPI package website*. PyPI. https://pypi.org/project/pydruglogics/

Szekeres, L. (2025). *Pipeline evaluation of PyDrugLogics*. https://github.com/druglogics/pydruglogics/wiki/Pipeline-Evaluation-of-PyDrugLogics

Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., Iersel, M. V., Lauffenburger, D. A., & Saez-Rodriguez, J. (2012). CellNOptR: A flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Systems Biology*, *6*(1), 133. https://doi.org/10.1186/1752-0509-6-133

Zobolas, J. (2020). *DrugLogics software documentation*. GitHub Pages. https://druglogics.github.io/druglogics-doc/