Targeting tissues via dynamic human systems modeling in generative design

Zachary R Fox
Oak Ridge National Laboratory
foxzr@ornl.gov

Nolan J English
Oak Ridge National Laboratory
Oak Ridge, TN
englishnj@ornl.gov

Belinda S Akpa
Oak Ridge National Laboratory
Oak Ridge, TN
akpabs@ornl.gov

Abstract

Drug discovery is a complex, costly process with high failure rates. A successful drug should bind to a target, be deliverable to an intended site of activity, and promote a desired pharmacological effect without causing toxicity. Typically, these factors are evaluated in series over the course of a pipeline where the number of candidates is reduced from a large initial pool. One promise of AI-driven discovery is the opportunity to evaluate multiple facets of drug performance in parallel. However, despite ML-driven advancements, current models for pharmacological property prediction are exclusively trained to predict molecular properties, ignoring important, dynamic biodistribution and bioactivity effects. Here, we present our progress towards incorporating quantitative systems physiology models into an ML-based molecular generation pipeline. Within a genetic algorithm, we include human-relevant physiologically based pharmacokinetic (PBPK) models. These PBPK models leverage properties that are predicted by a fine-tuned molecular language model. Together, these models will aid in capturing the mapping between molecules and therapeutic outcomes that is necessary to accelerate the drug discovery process.

Introduction

Due to an abundance of molecular property data and the enormous, complex design space that is largely inaccessible using traditional molecular modeling or high-throughput experimental drug discovery approaches, drug design is emerging as a key application area for machine learning. Machine learning (ML) approaches for design of molecular therapeutics largely fall into a handful of categories: property prediction, in which ML algorithms aim to predict molecular properties [1, 2, 3, 4]; hit expansion, in which ML algorithms aim to generate novel therapeutics based on an existing molecule [5]; synthesis prediction [6], and recently *de novo* drug design using generative modeling techniques [7, 8, 9]. Each of these categories has made use of modern ML algorithms (LLMs, Diffusion Models, GANs, and GNNs) and various molecular representations (fingerprints, graphs, sequences).

Many of these models are trained on molecular property data, and generative models are often used to develop molecules that have a high affinity to a particular target of interest. While these models are growing ever-more accurate, they often fail to properly account for the biological context of the drug within the human body. If generative design does not address the ability of a compound to accumulate

Preprint. Under review.

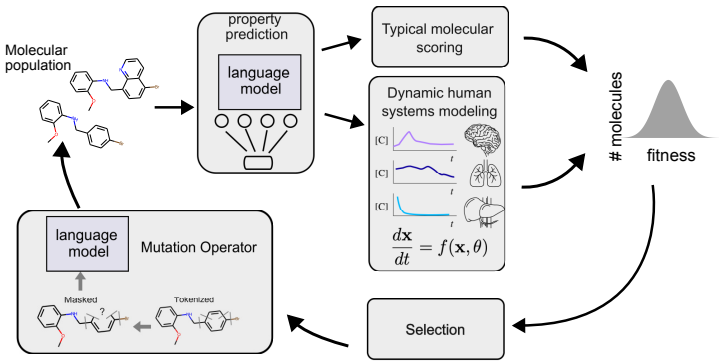

Figure 1: (A) Overview and distinction of potential drugs for typical property based optimization and detailed pharmacokinetic optimization. (B) PBPK-based generative drug design using language models, genetic algorithms, and systems models.

in a desired tissue and impact biological signaling pathways for sufficient time to be effective, these AI approaches will fall afoul of the same challenges that cause 90% of drug candidates entering clinical trials to fail [10]. Potently hitting a target protein is a necessary but insufficient condition for a molecule to be a drug.

Systems pharmacology models can predict the potential dynamic distribution and physiological effect of a candidate drug from the candidate's molecular properties [11]. These models are typically deployed fairly late in the drug discovery process, when molecules have been synthesized and their properties evaluated experimentally. Integrating human systems models into generative design could bring human physiology to bear on the earliest stages of discovery – permitting virtual screening of molecules for human dynamic distribution prior to drug synthesis and experimental evaluation. However, multiple practical challenges arise when trying to integrate systems pharmacology models into generative design workflows. We address two of them herein – namely: (1) systems models are computationally expensive and slow compared to neural network models, and (2) systems models have to rely on machine learning models to predict required, human-relevant input parameters; only very limited data exist to train these predictive models. Finally, we address the significance of constraining molecular design to molecules that are likely to be accessible via known, low-risk, and low-cost chemical synthesis pathways.

Here, we develop a framework for molecular generation that leverages detailed physiological models in the molecular scoring pipeline, shown in Fig. 1. As such, the main contribution of this work is, to the authors' knowledge, the first example of a generative molecular design approach that is informed using detailed, dynamic, pharmacokinetic modeling. We demonstrate quantitative differences in generated molecules that only consider target binding affinity.

Methods

Performant and rapid physiological modeling to assess tissue targeting Physiologically based pharmacokinetic models (PBPK) combine molecular data on drug candidates with prior knowledge of human physiology to predict organ-specific drug exposure and its consequences. These models are composed of a system of ordinary differential equations (ODEs) that represent the body as an assembly of organ compartments connected by circulating blood [12, 13, 14]. Each equation is a material balance that describes accumulation of drug in a particular organ (see [13]). Embedded in each equation are algebraic and kinetic relationships describing the biophysical and biochemical processes governing drug uptake at tissue extracellular and intracellular levels. Integrating these models into generative design workflows creates an opportunity to incorporate human-level outcomes directly into molecular design. However, relocating PBPK models to this early phase of discovery also creates new challenges.

In the context of generative molecular design, drug candidates will only exist as computational hypotheticals, because their properties must be predicted by using ML models that map chemical

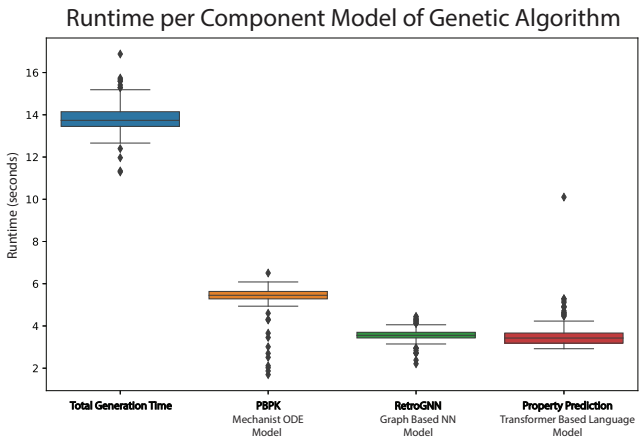

Figure 2: Execution time distributions for the LLM property prediction models, the PBPK Model solver, and Retrosynthetic score prediction by RetroGNN. Runtimes are given per generation a population of $\sim$2,000 molecules evaluated across 200 generations.

structure to molecular properties. Thus, the quality of the mechanistic PBPK predictions will depend on the performance of the ML models providing parameter inputs. In the case of properties like lipophilicity, which is a property that is independent of biology and for which there is abundant experimental data, training such models poses little concern. However, data reflecting molecular interactions with human biology are difficult to obtain, and their scarcity makes for lower predictive performance in ML models. Enabling the integration of systems and generative models for inverse design requires ML intermediates that make reasonable predictions from limited data.

A further challenge is the computational expense of mechanistic systems models. Solving the system of differential algebraic equations that make up a typical PBPK model can require run-times two orders of magnitude longer than running a drug molecule through a machine learning model. To mitigate this, we implemented our PBPK model in Julia, taking advantage of both `ModelingToolkit.jl` and `Symbolics.jl` [15], along with solvers from `OrdinaryDiffEq.jl` [16]. After compilation, the model had a solution time of 1.3 ms per drug molecule. This is nearly 100 times faster than the 123 ms per drug required by our `MATLAB` prototype. While some of this speed-up can be attributed to the inherent optimizations in Julia, the majority of the time savings comes from the symbolic simplification enabled by `ModelingToolkit.jl`. After applying these optimizations, we were able to reduce our system of differential-algebraic equations from approximately 130 equations down to just 14. Consequently, predicting the time course of a drug candidate is now on par with both the graph based and language based machine learning models as shown in Fig. 2. Further details are given in Appendix C. These advances mean can now invoke dynamic human-level evaluative criteria and thus design for the physiologically-relevant properties that make a molecule a drug.

Language model for generating molecules Molecular representations for generating molecules are a topic of ongoing investigation. In particular, molecular string representations such as SMILES [17] have shown much promise, as they enable the straightforward application of natural language programming (NLP) tools [18]. A given SMILES string is converted into a set of tokens, which uses frequencies of subsets of these representations to build a vocabulary [19, 20]. Tokenized molecules are then used to train a masked language model (MLM), which reconstructs the original molecular string after random tokens have been omitted. Importantly, this process is unsupervised, enabling the use of large, unlabeled molecular datasets, such as the Enamine REAL database [21]. After training, MLMs can be used to generate molecules by first randomly masking a portion of the tokenized molecules and then sampling from the rank-sorted predicted tokens. In this work, the LM serves as a mutation operator within a genetic algorithm (GA), to generate plausible and diverse sets of molecules [22, 23, 24].

Fine-tuning language models for physiological property prediction In addition to generative modeling, molecular embeddings from the LM can be used as an informative space from which other chemical properties are predicted. Recently, affinity was predicted by fine tuning a pretrained language model for drug-like molecules along with a protein language model`ProtBERT` [22, 25]. We

used the embeddings from the LM described above to fine tune three different models for molecular property prediction (see Appendix B.1 and B.2).

Our PBPK model required prediction of six molecular properties. Four properties, acidic and basic pKa constants, fraction unbound in plasma (fub) and lipophilicity (LogD) were available from the Lombardo dataset [26]. For these four properties, we fine-tuned the pretrained LM on a train/validation/test splits of 686/86/86 molecules (see Fig. A2). For drug efficacy, we fine-tuned the model on activity metrics (IC50) from the PostEra dataset describing inhibitors of the SARS-CoV2 main protease (MPro) [27]. Finally, we inferred intrinsic clearance values from the Lombardo datasets by assuming dominant hepatic clearance, limited to a maximum dictated by hepatic blood flow.

Retrosynthesis-informed screening for synthesizable molecules using RetroGNN In drug screening, synthesizability is key and can be evaluated in two ways: complexity-based and retrosynthesis-based methods [28]. Complexity-based methods focus on structural features like stereocenters, while retrosynthesis methods reverse engineer feasible synthetic routes and offer cost and yield insights. Though valuable, retrosynthesis methods are computationally demanding, requiring large compound libraries and significant computing resources. Consequently, complexity-based methods have been the default for estimating synthesizability. Recent machine learning advances, like RetroGNN built on Chemprop [6, 29], now allow for efficient approximation of retrosynthetic accessibility scores, originally generated by tools like MoleculeOne and Aizynthfinder [6, 30]. By incorporating approximate retrosynthetic analysis models into our workflow, we increase the likelihood of identifying molecules that are not just potent, deliverable candidates, but are also credibly synthesizable.

Genetic algorithms to optimize molecules The optimization of molecular properties is a challenging, high-dimensional problem. For such global search problems, genetic algorithms have been successful in numerous domains [31, 32, 33]. In the world of molecular design, previous works have used physically inspired heuristics such as mutating atoms and bonds [34, 35, 36], fragment based rearrangement, [37, 38, 36, 39, 40], and other handcrafted mutation rules [34] in the context of evolutionary algorithms. Recently, language models have been used as the mutation operator in genetic algorithms to generate new molecules [22, 23] within the population (see Appendix A).

Results

To demonstrate the significance of using dynamic pharmacokinetic models in generative molecular design, in this preliminary work, we demonstrate how an initial set of molecules can be evolved *in silico* to improve the practical efficacy of a SARS-CoV2 protease inhibitor.

We used molecules from the PostEra database [27] as the initial seed for our genetic algorithm. In each generation of the algorithm, 25% of the tokenized SMILES strings from the previous generation were masked. These masks were then filled by a trained masked language model (MLM), serving as the mutation step within the genetic algorithm. Subsequently, each member of the population was evaluated to determine their fitness , (Fig. 1), with fitness defined by a scaled average of pharmacokinetic, molecular, and retrosynthetic properties see B.2. The top 5,000 members, according to this fitness, were carried forward into the next generation. Our algorithm successfully evolved the molecule population to improve synthetic accessibility and increase drug exposure at the site of activity, as shown in Fig. 3A,A4.

Next, we compared the outcomes associated with two different fitness functions: one determined solely on pIC50 (potency), and another that considers pIC50, retrosynthetic accessibility, and pharmacokinetic properties (Appendix B.2). The latter improved both retrosynthetic accessibility and the drug's unbound concentration in the desired lung sub-compartment. (Fig. 3B, A5).

Discussion

We have demonstrated of the impact of dynamic PBPK modeling on optimization of a protease inhibitor. Notably, including retrosynthetic and PBPK metrics in our fitness functions led to small changes in our seed molecules, but significant predicted improvements for drug disposition within a target tissue, Fig.3C. A limitation of this work is that it relies on model predictions of several drug properties which may or may not be accurate. Furthermore, these errors may be compounded when passed through the PBPK model. Future work will refine our property prediction models and quantify uncertainty in these estimates, and investigate the sensitivity of the PBPK model to

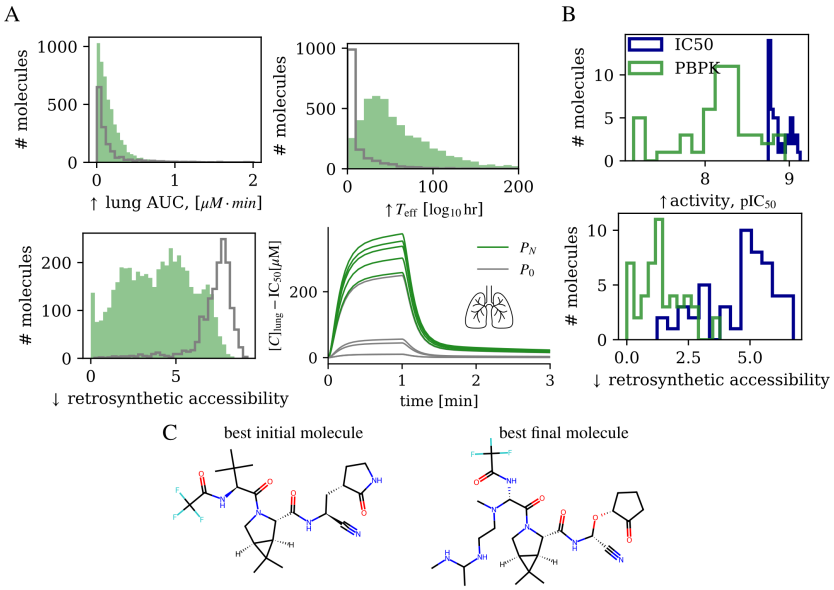

Figure 3: (A) Property histograms for the initial (gray) and final (green) molecule populations, with PBPK properties having been included in the fitness function for lung intracellular AUC, T_{eff} (time above IC50), and retrosynthetic accessibility. The bottom right figure shows the predicted concentration-time curves for individual molecules in the intracellular space of lung tissue, expressed as $[C_{\text{lung}}]$ minus the predicted IC_{50}. Curves are shown for the top 5 molecules in the initial (gray) and final (green) populations. (B) Distribution of activity (pIC_{50}) and retrosynthetic accessibility for the population of molecules obtained by searching chemical space using only activity (blue) or using a model that includes activity and PBPK parameters (green). (C) The best molecule (by fitness) in the initial and final populations.

changes in molecular properties. We will also explore alternative fitness functions using multiple weighting schemes of the various terms in the function. Another avenue for further optimization is the development of end-to-end differentiable models, encompassing property prediction through PBPK modeling, to further optimize molecular candidates.

Acknowledgments and Disclosure of Funding

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). The research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration, and by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy (Award 10493 to BSA). NJE and BSA also acknowledge the Accelerating Therapeutics for Opportunities in Medicine (ATOM) Consortium. “Liver", “Lungs", “Brain" by iconmu from Noun Project.

References

[1] Benedek Fabian, Thomas Edlich, Helena Gaspar, Marwin H. S. Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *ArXiv*, abs/2011.13230, 2020.

[2] Nereida Rodríguez-Fernández Francisco Cedrón Francisco J. Novoa Adrian Carballal Victor Maojo Alejandro Pazos Carlos Fernandez-Lozano Paula Carracedo-Reboredo, Jose Liñares-Blanco. A review on machine learning approaches and trends in drug discovery. *Computational and Structural Biotechnology Journal*, 19:4538–4558, 2021.

[3] Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. Molecular representations in AI-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12(1):56, September 2020.

[4] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022.

[5] Daiki Erikawa, Nobuaki Yasuo, and Masakazu Sekijima. MERMAID: an open source automated hit-to-lead method based on deep reinforcement learning. *Journal of Cheminformatics*, 13(1):94, November 2021.

[6] Cheng-Hao Liu, Maksym Korablyov, Stanisław Jastrzębski, Paweł Włodarczyk-Pruszyński, Yoshua Bengio, and Marwin Segler. Retrognn: Fast estimation of synthesizability for virtual screening and de novo design by learning from slow retrosynthesis software. *Journal of Chemical Information and Modeling*, 62(10):2293–2300, 2022. PMID: 35452226.

[7] Vidya Niranjan, Akshay Uttarkar, Ananya Ramakrishnan, Anagha Muralidharan, Abhay Shashidhara, Anushri Acharya, Avila Tarani, and Jitendra Kumar. De Novo Design of Anti-COVID Drugs Using Machine Learning-Based Equivariant Diffusion Model Targeting the Spike Protein. *Current Issues in Molecular Biology*, 45(5):4261–4284, May 2023.

[8] Mingyang Wang, Zhe Wang, Huiyong Sun, Jike Wang, Chao Shen, Gaoqi Weng, Xin Chai, Honglin Li, Dongsheng Cao, and Tingjun Hou. Deep learning approaches for de novo drug design: An overview. *Current Opinion in Structural Biology*, 72:135–144, February 2022.

[9] Maryam Abbasi, Beatriz P. Santos, Tiago C. Pereira, Raul Sofia, Nelson R. C. Monteiro, Carlos J. V. Simões, Rui M. M. Brito, Bernardete Ribeiro, José L. Oliveira, and Joel P. Arrais. Designing optimized drug candidates with Generative Adversarial Network. *Journal of Cheminformatics*, 14(1):40, June 2022.

[10] Asher Mullard. Parsing clinical success rates. *Nature Reviews Drug Discovery*, 15(7):447–447, July 2016. Number: 7 Publisher: Nature Publishing Group.

[11] Simone Q. Pantaleão, Philipe O. Fernandes, José Eduardo Gonçalves, Vinícius G. Maltarollo, and Kathia Maria Honorio. Recent advances in the prediction of pharmacokinetics properties in drug design studies: A review. *ChemMedChem*, 17(1):e202100542, 2022.

[12] Malcolm Rowland, Carl Peck, and Geoffrey Tucker. Physiologically-based pharmacokinetics in drug development and regulatory science. *Annual Review of Pharmacology and Toxicology*, 51:45–73, 2011.

[13] Ilin Kuo and Belinda S Akpa. Validity of the lipid sink as a mechanism for the reversal of local anesthetic systemic toxicity: a physiologically based pharmacokinetic model study. *Anesthesiology*, 118(6):1350–1361, 2013.

[14] Patrick Poulin, Rhys D O Jones, Hannah M Jones, Christopher R Gibson, Malcolm Rowland, Jenny Y Chien, Barbara J Ring, Kimberly K Adkison, M Sherry Ku, Handan He, Ragini Vuppugalla, Punit Marathe, Volker Fischer, Sandeep Dutta, Vikash K Sinha, Thorir Björnsson, Thierry Lavé, and James W T Yates. Phrma cpcdc initiative on predictive models of human pharmacokinetics, part 5: prediction of plasma concentration-time profiles in human by using the physiologically-based pharmacokinetic modeling approach. *Journal of Pharmaceutical Sciences*, 100(10):4127–4157, 2011.

[15] Yingbo Ma, Shashi Gowda, Ranjan Anantharaman, Chris Laughman, Viral Shah, and Chris Rackauckas. Modelingtoolkit: A composable graph transformation system for equation-based modeling, 2021.

[16] Christopher Rackauckas and Qing Nie. Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1):15, 2017.

[17] David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28:31–36, 1998.

[18] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm):4171–4186, 2019.

[19] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, 2012.

[20] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. pages 1–23, 2016.

[21] Enamine *REAL* Database. `https://enamine.net/compound-collections/real-compounds/real-database`. Accessed: 2020-04-01 through `https://virtual-flow.org/`.

[22] Andrew E. Blanchard, John Gounley, Debsindhu Bhowmik, Mayanka Chandra Shekar, Isaac Lyngaas, Shang Gao, Junqi Yin, Aristeidis Tsaris, Feiyi Wang, and Jens Glaser. Language Models for the Prediction of SARS-CoV-2 Inhibitors.

[23] Andrew E. Blanchard, Mayanka Chandra Shekar, Shang Gao, John Gounley, Isaac Lyngaas, Jens Glaser, and Debsindhu Bhowmik. Automating Genetic Algorithm Mutations for Molecules Using a Masked Language Model. *IEEE Transactions on Evolutionary Computation*, 2022.

[24] Andrew E. Blanchard, Debsindhu Bhowmik, Zachary Fox, John Gounley, Jens Glaser, Belinda S. Akpa, and Stephan Irle. Adaptive language model training for molecular design. *Journal of Cheminformatics*, 15(1):59, June 2023.

[25] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

[26] Franco Lombardo, Giuliano Berellini, and R. Scott Obach. Trend Analysis of a Database of Intravenous Pharmacokinetic Parameters in Humans for 1352 Drug Compounds. *Drug Metabolism and Disposition: The Biological Fate of Chemicals*, 46(11):1466–1477, November 2018.

[27] John Chodera, Alpha A. Lee, Nir London, and Frank von Delft. Crowdsourcing drug discovery for pandemics. *Nature Chemistry*, 12(7):581–581, July 2020. Number: 7 Publisher: Nature Publishing Group.

[28] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):1–11, 2009.

[29] Esther Heid, Kevin Greenman, Yunsie Chung, Shih-Cheng Li, David Graff, Florence Vermeire, Haoyang Wu, William Green, and Charles McGill. *Chemprop: A machine learning package for chemical property prediction*, 2023.

[30] Samuel Genheden, Amol Thakkar, Veronika Chadimová, Jean-Louis Reymond, Ola Engkvist, and Esben Bjerrum. Aizynthfinder: A fast, robust and flexible open-source software for retrosynthetic planning. *Journal of Cheminformatics*, 12(1), 2020.

[31] Nathan Brown, Marco Fiscato, Marwin H.S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking Models for de Novo Molecular Design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019.

[32] Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*. Springer, 2015.

[33] Gregory S Hornby, Jason D Lohn, and Derek S Linden. Computer-Automated Evolution of an X-Band Antenna for NASA's Space Technology 5 Mission. *Evolutionary Computation*, 19(1):1–23, 2011.

[34] Aaron M. Virshup, Julia Contreras-García, Peter Wipf, Weitao Yang, and David N. Beratan. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *Journal of the American Chemical Society*, 135(19):7296–7303, 2013.

[35] Jan H. Jensen. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019.

[36] Eric Wubbo Lameijer, Joost N. Kok, Thomas Bäck, and Ad P. Ijzerman. The molecule evoluator. An interactive evolutionary algorithm for the design of drug-like molecules. *Journal of Chemical Information and Modeling*, 46(2):545–552, 2006.

[37] Christos A. Nicolaou, Joannis Apostolakis, and Costas S. Pattichis. De novo drug design using multiobjective evolutionary graphs. *Journal of Chemical Information and Modeling*, 49(2):295–307, 2009.

[38] Nathan Brown, Ben McKay, François Gilardoni, and Johann Gasteiger. A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *Journal of Chemical Information and Computer Sciences*, 44(3):1079–1087, 2004.

[39] Eric Wubbo Lameijer, Joost N. Kok, Thomas Back, and Ad P. Ijzerman. Mining a chemical database for fragment co-occurrence: Discovery of "chemical clichés". *Journal of Chemical Information and Modeling*, 46(2):553–562, 2006.

[40] Gisbert Schneider, Man Ling Lee, Martin Stahl, and Petra Schneider. De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *Journal of Computer-Aided Molecular Design*, 14(5):487–494, 2000.

Appendix

A Genetic Algorithm Pseudocode

Algorithm 1 Pseudocode Genetic Algorithm for Molecule Optimization

Initialize: Population of molecules
Population size = N_{pop}
number of generations = N_{gen}
mutation rate = 25%
for gen = 1 to generations **do**
  Mutation Step:
  while current_pop < max_pop **do**
    Mask percentage of SMILES sequence by token in population
    Use language model to replace masks
    if generated molecules are invalid (syntactically or semantically) **then**
      Discard invalid molecules
    end if
    Duplicate reduced population and append to previous set
  end while
  Property Prediction:
  for all molecules in population **do**
    Predict properties using pretrained language model
    if pIC50 < 7 **then**
      Discard molecule
    else
      Calculate concentration-time curve metrics with PBPK model
      Predict retrosynthetic accessibility via RetroGNN
    end if
  end for
  Normalization:
  Scale values against initial population via z-normalization
  Selection:
  Evaluate population based on Experimental fitness function
  Select members for the next generation
end for

B Data transformations

B.1 Multitask physiological property prediction model

The multitask fine-tuned Transformer model was used to simultaneously predict multiple properties, where the loss function includes contributions from each of the five properties: MW, PKa acidic, pKa basic, fraction unbound in plasma (fub) and lipophilicity (LogD). Each property is scaled and taken into consideration in a standard mean-squared error loss function

$$\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{5} \left(f(y_k; \boldsymbol{\alpha}_k) - \mathrm{NN}_k(\mathbf{x}; \theta) \right)^2 . \tag{1}$$

The invertible function $f(\cdot; \boldsymbol{\alpha}_k)$ transforms a given vector $\mathbf{y}$ such that each entry is on a comparable scale, such that Eq. 1 does not over/under weigh individual properties. At inference, the inverse function $f^{-1}(\tilde{\mathbf{y}}, \boldsymbol{\alpha}_k)$ maps the neural network predicted vector $\tilde{\mathbf{y}} = \mathrm{NN}_k(\mathbf{x}; \theta)$ back to the appropriate scale for each property. Fine tuning results for the model are given in Table A.1

Each of MW, PKa acidic, pKa basic and LogD were all transformed according to a simple scaling $\tilde{y}_k = \frac{y_k - \mu_k}{\sigma_k}$, where the μ_k and σ_k are the mean and standard deviation of y_k over the entire dataset.

	pKa acidic	pKa basic	fub	LogD
R^2	0.763	0.617	0.473	0.554
MAE	1.73	1.81	0.667	0.879
RMSE	2.24	2.32	0.834	1.28

Table A.1: multitask fine tuned model results on the Lombardo-Obach dataset

For fraction unbound, prior to the normalization mentioned above, we first take the logarithmic transformation

$$\tilde{y}_k = \log \frac{(1 - y_k)}{y_k}. \tag{2}$$

B.2 Normalization of properties for molecular scoring

Within our genetic algorithm, the top N_{pop} molecules are selected to remain in the population given their fitness. We define fitness of a molecule as the weighted average of different properties of the molecule, or of scalar properties of the PBPK model solutions. When multiple properties are used to compute the fitness, we consider an average of the prescribed values. The properties which were considered in the fitness are given below:

	T_{eff}	AUC_{lung}	pIC_{50}	retrosynthetic accessibility
PBPK	✓	✓	✓	✓
pIC_{50}	-	-	✓	-

T_{eff} was calculated by simulating the PBPK model for a given molecule and then determining the duration of time over which the unbound drug concentration in the lung intracellular space exceeded the half maximal inhibitory concentration, IC_{50}. Similarly, AUC_{lung} is the unbound drug concentration in the lung intracellular space integrated over the considered treatment time. pIC_{50} is predicted by the fine-tuned LM. Retrosynthetic accessibility comes from `RetroGNN` [6].

To combine multiple properties for scoring, we needed to standardize their values. Each of the properties above was computed for the initial molecular population, which is identical for each experiment. For T_{eff}, $\log_{10} \text{AUC}_{\text{lung}}$, and pIC_{50}, we used a simple standardization scheme, such that the mean value is near 0.5, and most of the values fall within the range [0,1], i.e.

$$\tilde{s}_k = \frac{s_k - \mu_k}{6\sigma_k} + 0.5 \tag{3}$$

where s_k corresponds to the original score and $\tilde{s}_k$ to the transformed score.

C Accelerating physiologically based pharmacokinetic model (PBPK) predictions

Pharmacokinetic models (PK) are typically composed of systems of equations describing the flow of a compound through connected compartments representing tissues and vascular spaces within the body. One can vary the level of granularity employed (*e.g.*, number of explicitly depicted vs. lumped compartments or the number of explicitly depicted tissue sub-compartments such as intracellular and extracellular space, or even intracellular sub-spaces). Depending on the level of granularity, these models can have as few as a dozen equations to hundreds of equations describing the disposition of a compound into sub-compartments of all included tissues. As an example of the latter type of model, our PBPK model includes 14 compartments (venous, arterial, and 12 organ compartments). Each organ compartment is further subdivided into vascular, tissue intracellular and tissue extracellular space. While a treatment of this model and its assumptions is beyond the scope of this work, this systems model requires the solution of over a hundred equations. Given the iterative nature of solving systems of ordinary differential equations, even optimized code can be time consuming to execute.

To accelerate mechanistic systems models, two primary approaches are generally considered: optimizing the code execution and pre-optimizing the mathematical models from which the code originates.

For the former, techniques like memory optimization, vectorization, and parallelization are increasingly accessible, due to their integration into modern programming languages. Julia is particularly noteworthy in this context. It offers intelligent memory management and employs stringent type inference, allowing for type-specific optimizations that lead to faster execution and compilation times.

In contrast, pre-optimizing mathematical models usually demands specialized knowledge, particularly for reducing systems of equations through simplification or approximation. While code optimization can often be automated, equation optimization typically remains a manual process. Conveniently, Symbolic Algebra Systems (SAS) offer a route to automation in this area. Without delving into the intricate details of how SAS approaches use symbolic representation to manipulate mathematical expressions, it's worth noting that numerous SAS implementations exist in Julia. These allow for the automated optimization of complex systems of equations.

In our project, we chose to implement our PBPK model in Julia, taking advantage of both `ModelingToolkit.jl` and `Symbolics.jl` [15], along with solvers from `OrdinaryDiffEq.jl` [16]. After compilation, the model had a solution time of 1.3 ms per drug molecule. This is nearly 100 times faster than the 123 ms per drug required by our MATLAB prototype. While some of this speed-up can be attributed to the inherent optimizations in Julia, the majority of the time savings comes from the symbolic simplification enabled by `ModelingToolkit.jl`. After applying these SAS optimizations, we were able to reduce our system of differential-algebraic equations from approximately 130 equations down to just 14.

To validate these performance improvements, 200 generations of a 20,000 molecule population were processed through our genetic algorithm (Each generation is shown individually in A1. These experiments were conducted on an AMD Ryzen 9 7950X 16-Core CPU and NVIDIA GeForce RTX 3090 Ti with 128GB RAM.

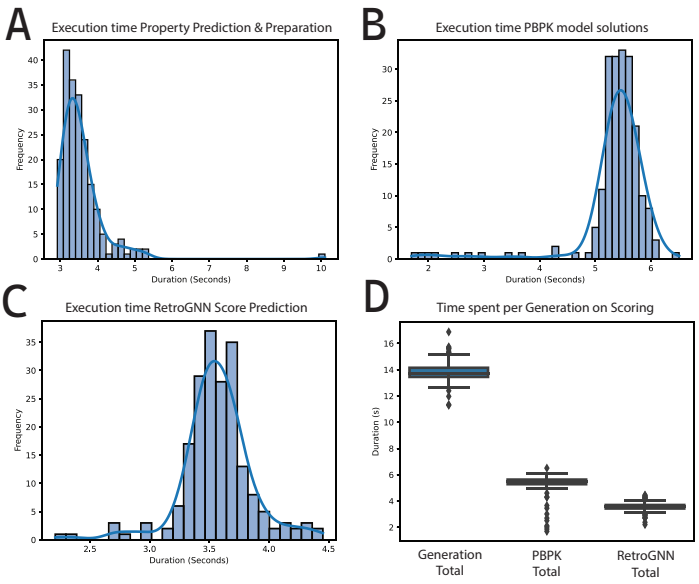

Figure A1: (A,B,C) Execution time distributions for (A) the LLM property prediction models, (B) the PBPK Model solver, and (C) Retrosynthetic score prediction by RetroGNN. Times are expressed in seconds, evaluated across 200 generations, with $\sim$2000 molecules per generation. (D) On average, of the total Execution time, via the PBPK model occupies 34% of the execution time per generation.

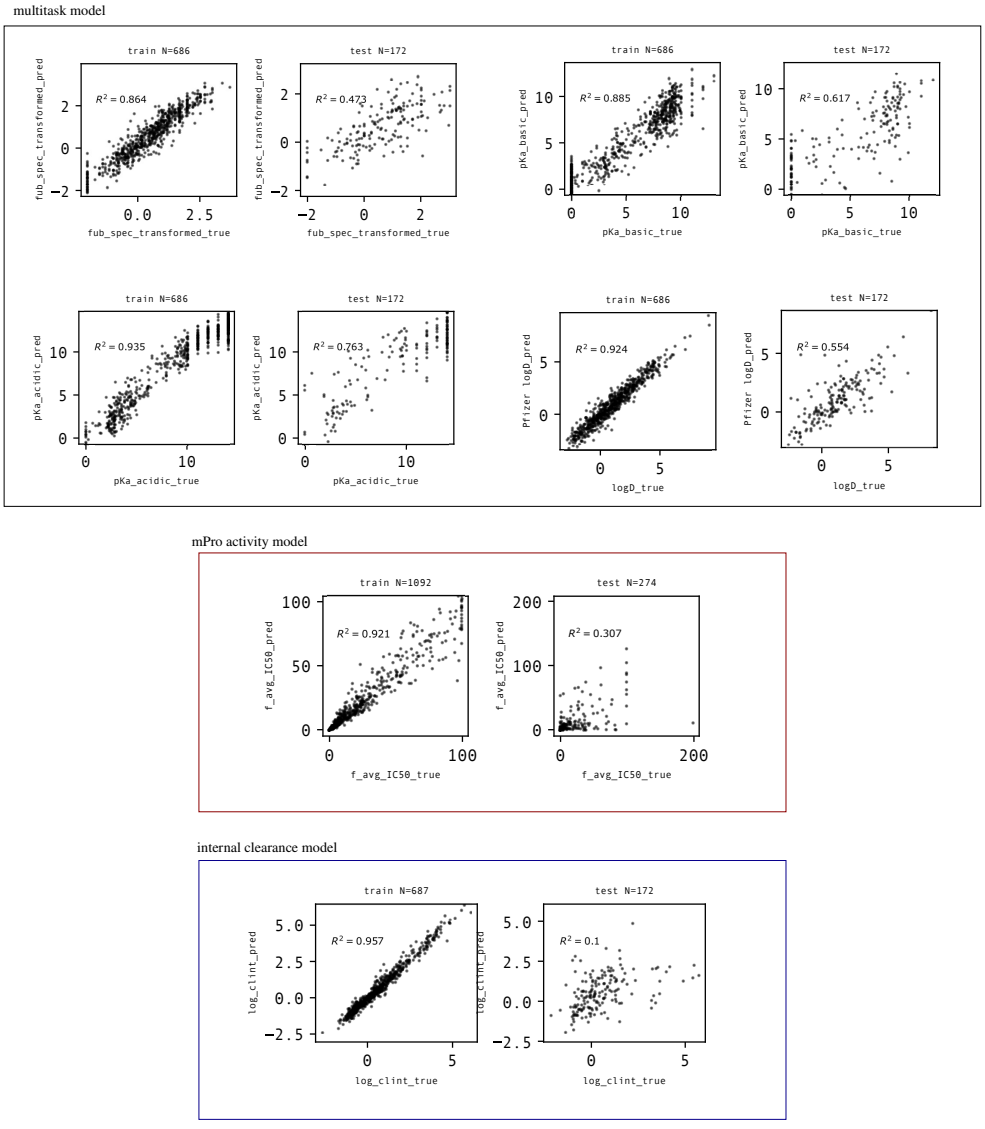

Figure A2: Parity plots for each of the fine-tuned language models.

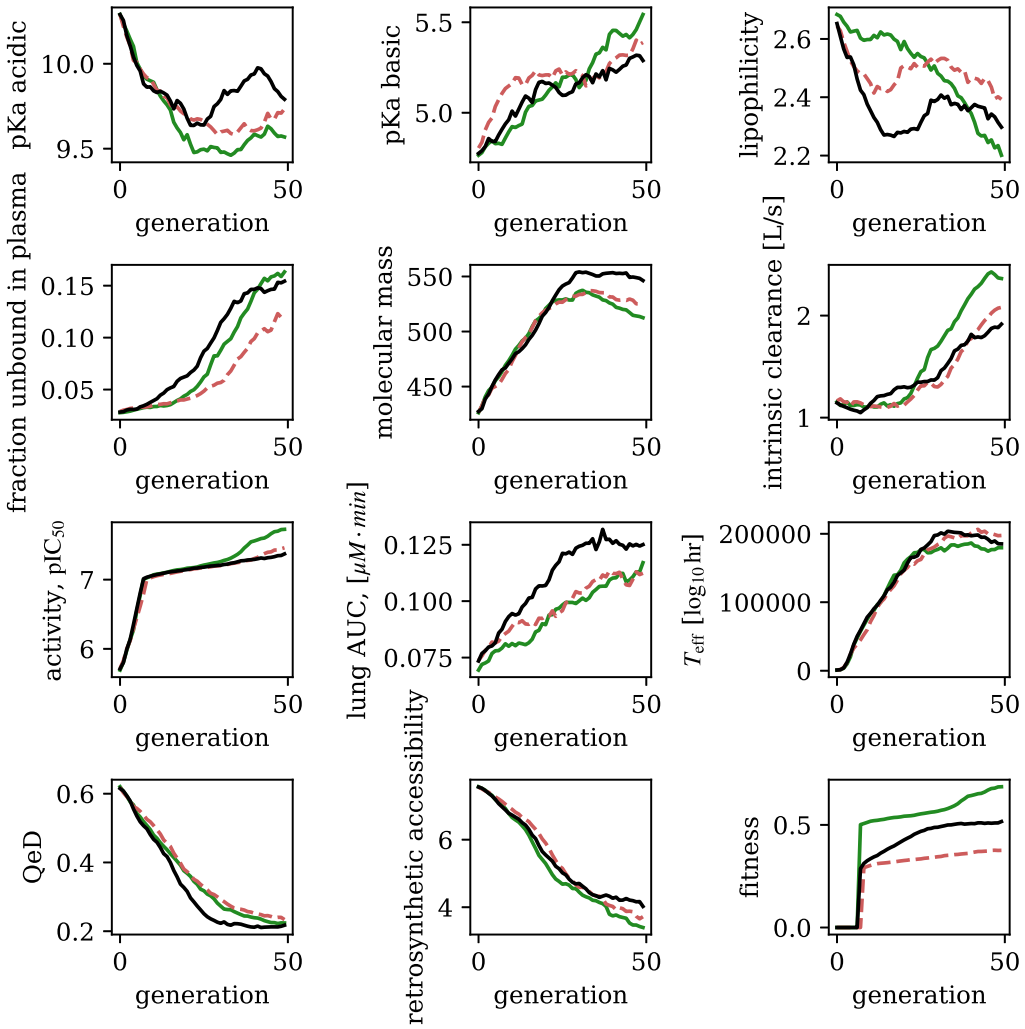

Figure A3: Median values of each property for molecules in the population at each iteration. The fitness function for the black trace includes pIC50, PBPK metrics, and RetroGNN scores. The fitness function for the red trace includes pIC50 and PBPK metrics. The fitness function for the green trace only includes pIC50.

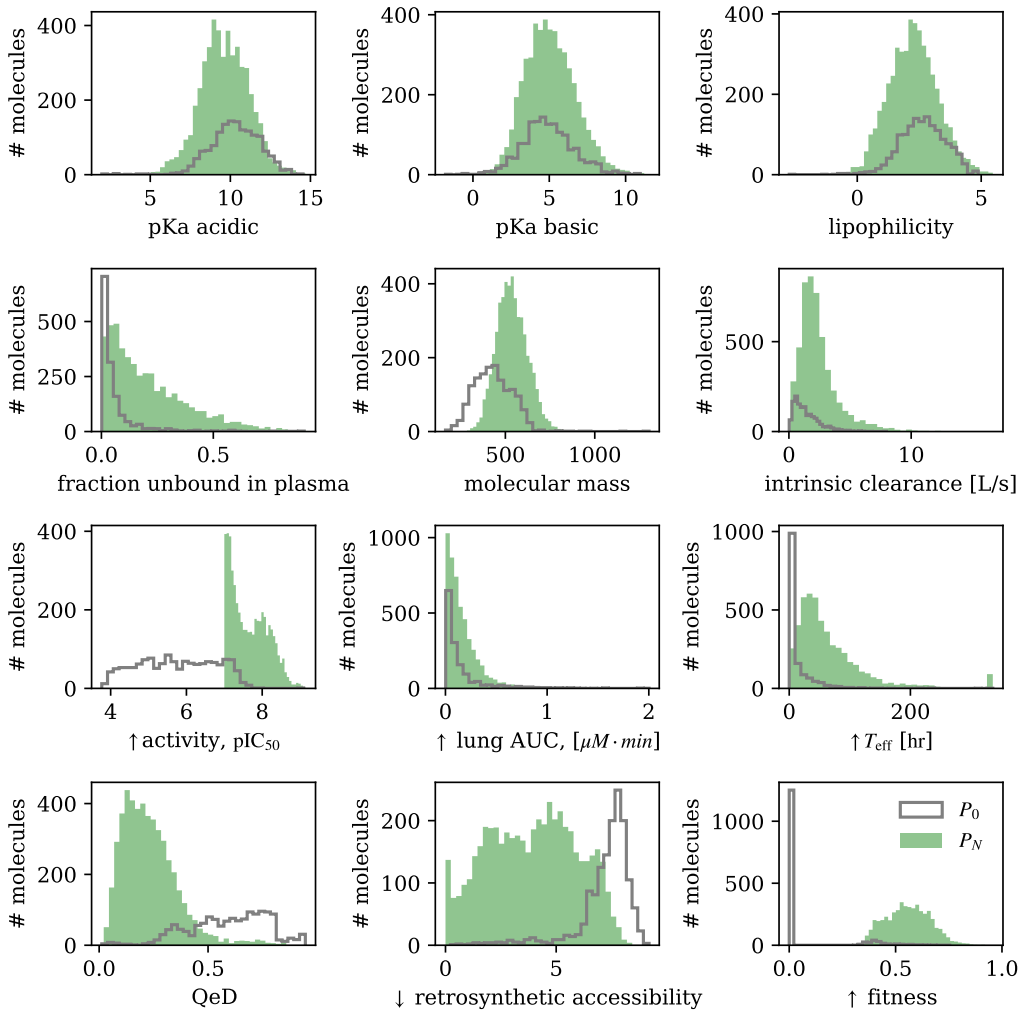

Figure A4: Distributions of the molecular properties in the initial population (gray) and final population (green). Arrows next to properties indicate direction of contribution to higher fitness.

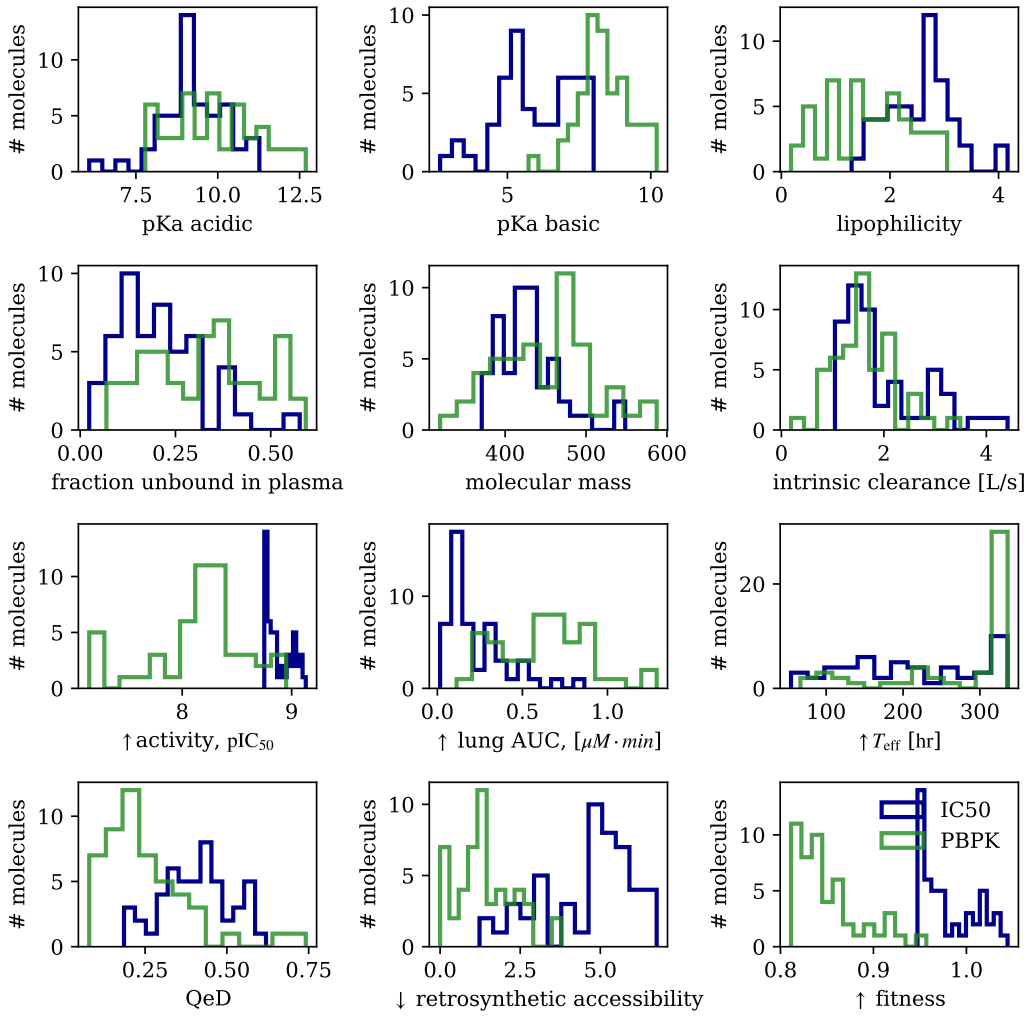

Figure A5: Distributions of the molecular properties of the top 50 candidates using only pIC50 as fitness (blue) and including both PBPK and RetroGNN (green). Arrows next to properties indicate direction of contribution to higher fitness.