Genetic and Evolutionary Computation Conference (GECCO '25)
Symbolic Regression Workshop
July 14-18, 2025, Málaga, Spain

Call for Action: Towards the Next Generation of Symbolic Regression Benchmark

Guilherme Seidyo Imai Aldeia
guilherme.aldeia@ufabc.edu.br
Federal University of ABC
Santo André, São Paulo, BR

Hengzhe Zhang
hengzhe.zhang@ecs.vuw.ac.nz
Victoria University of Wellington
Wellington, NZ

Geoffrey Bomarito
geoffrey.f.bomarito@nasa.gov
NASA Langley Research Center
Hampton, Virginia, USA

Miles Cranmer
mc2473@cam.ac.uk
University of Cambridge
Cambridge, UK

Alcides Fonseca
me@alcidesfonseca.com
LASIGE, Faculdade de Ciências da Universidade de Lisboa
Lisboa, PT

Bogdan Burlacu
University of Applied Sciences Upper Austria
Upper Austria, AT

William G. La Cava
william.lacava@childrens.harvard.edu
Computational Health Informatics Program
Boston Children's Hospital
Harvard Medical School
Boston, Massachusetts, USA

Fabrício Olivetti de França
folivetti@ufabc.edu.br
Federal University of ABC
Santo André, São Paulo, BR

SRBench

Framework for evaluating Symbolic Regression (SR) algorithms.

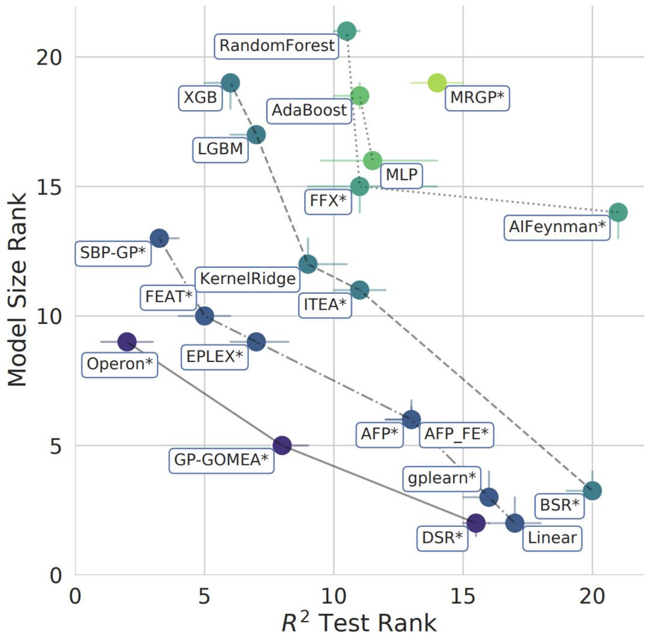


Figure 2: Pareto plot comparing the rankings of SR methods in terms of model size and R^2 score on the black-box problems. Points denote median rankings and the bars denote 95% confidence intervals. Connecting lines and color denote Pareto dominance rankings.

La Cava, W., Burlacu, B., Virgolin, M., Kommenda, M., Orzechowski, P., de França, F. O., ... & Moore, J. H. (2021). Contemporary symbolic regression methods and their relative performance. Advances in neural information processing systems, 2021(DB1), 1.

Our work

A call for action before the next official release of SRBench

25 methods evaluated (previously 14)

30 independent runs for each experiment (previously 10)

Subset of 12 datasets from the black-box problems (previously 122)

Replaced ground-truth with phenomenological and first principles datasets

Changed aggregated visualization to a new report plots

Experimental setup

Containers, 10GB RAM, single core, max runtime of 7 hours

Table 1: Metadata for each benchmark track. Dataset names match their corresponding PMLB entries

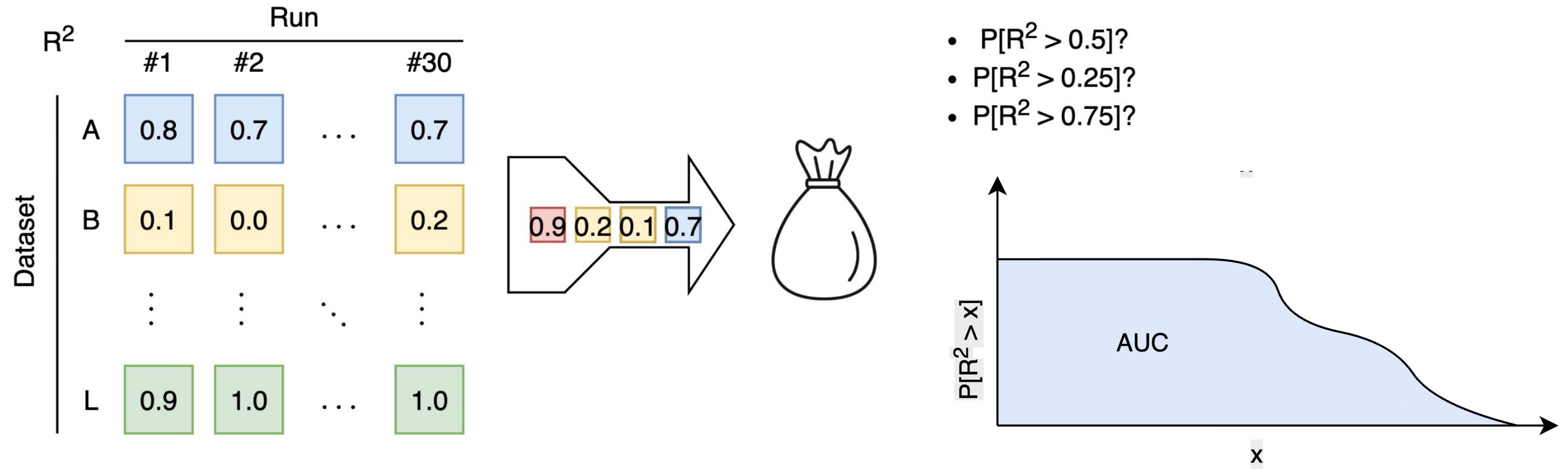
Black-box	# Rows	# Cols	Codomain
1028_SWD	1000	11	$\mathbb{Z}^+$
1089_USCrime	47	14	$\mathbb{Z}^+$
1193_BNG_lowbwt	31104	10	$\mathbb{R}^+$
1199_BNG_echoMonths	17496	10	$\mathbb{R}$
192_vineyard	52	3	$\mathbb{R}^+$
210_cloud	108	6	$\mathbb{R}^+$
522_pm10	500	8	$\mathbb{R}^+$
557_analcatdata_apnea1	475	4	$\mathbb{Z}^+$
579_fri_c0_250_5	250	6	$\mathbb{R}$
606_fri_c2_1000_10	1000	11	$\mathbb{R}$
650_fri_c0_500_50	500	51	$\mathbb{R}$
678_visualizing_environmental	111	4	$\mathbb{R}^+$
Phenomenological & first-principles	**# Rows**	**# Cols**	**Data source**
first_principles_absorption	14	2	Russeil et al. [53]
first_principles_bode	8	2	Bonnet [5]
first_principles_hubble	32	2	Hubble [26]
first_principles_ideal_gas	30	4	Generated, 10% noise
first_principles_kepler	6	2	Kepler [33]
first_principles_leavitt	26	2	Leavitt and Pickering [43]
first_principles_newton	30	4	Generated, 10% noise
first_principles_planck	100	3	Generated, 10% noise
first_principles_rydberg	50	3	Generated, 1% noise
first_principles_schechter	27	2	Generated, 20% noise
first_principles_supernovae_zr	236	2	Russeil et al. [53]
first_principles_tully_fisher	18	2	Tully and Fisher [59]

Table 2: Algorithms evaluated, their original references, and relevant characteristics pertinent to benchmarking.

Algorithm	Const. Opt.	Time limit	Multiple solutions	Runs on	Language	Description
AFP [56]	✗	✓	✗	CPU	C++	Age-fitness Pareto (AFP) optimization, meaning model age is used as an objective, with constants randomly changed
AFP_fe	✗	✓	✗	CPU	C++	AFP with co-evolved fitness estimates
AFP_ehc [39]	✓	✓	✗	CPU	C++	AFP with epigenetic hill climbing for constants optimization as local search
Bingo [51]	✓	✓	PF	CPU	Python	Evolves acyclic graphs with non-linear optimization, using islands for managing parallel populations
Brush [9]	✓	✓	PF	CPU	C++	GP with multi-armed bandits for controlling search space exploration
BSR [28]	✗	✓	✗	CPU	Python	Bayesian model with priors for operators and coefficients is used to sample expression trees
E2E [29]	✗	✗	✗	GPU	Python	Generator using pre-trained transformers, using BFGS and subsampling for tuning parameters
EPLEX [40]	✗	✓	✗	CPU	C++	GP with ϵ-lexicase parent selection
EQL [54]	✗	✗	✗	CPU	Python	Shallow neural network using mathematical operators as activation functions, and performs a pruning to refine the network to an expression
FEAT [8]	✓	✓	PF	CPU	C++	GP algorithm with ϵ-lexicase selection and linear combination of expressions using L1-OLS
FFX [47]	✓	✗	✗	CPU	Python	Non-evolutionary, deterministic approach, that generates a set of base functions and fits a regularized OLS to combine them
Genetic Engine [20]	✗	✓	✓	CPU	Python	GP using Context-Free Grammars to guide the generation process in an efficient manner
GPGomea [61]	✓	✗	✓	CPU	C++	GP with linkage learning used to propagate patterns and avoid their disruption
GPlearn	✗	✗	✓	CPU	Python	Canonical GP implementation
GPZGD [18]	✓	✓	✗	CPU	C	GP with Z-score standardization and stochastic gradient descent for parameter optimization
ITEA [12]	✓	✗	✗	CPU	Haskell	Mutation-based algorithm with constrained representation and OLS parameter optimization
NeSymRes [4]	✓	✓	✗	GPU	Python	Pre-trained encoder-decoders generate equation skeletons, optimized with non-linear optimization
Operon [34]	✓	✓	PF	CPU	C++	GP algorithm with weighted terminals and non-linear OLS parameter optimization
Ps-Tree [64]	✗	✗	✗	CPU	Python	GP algorithm that evolves Piecewise trees with SR expressions as leaves
PySR [10]	✓	✓	✗	CPU	Julia	Evolve-simplify-optimize loop with islands to manage parallel populations
Qlattice [6]	✓	✓	✗	CPU	Python	Uses a learned probability distribution updated over iterations to sample expressions, with parameter optimization. Closed source software
Rils-rols [31]	✓	✓	✗	CPU	C++	Iterative generation of perturbations and parameter optimization with OLS and local search selection of next candidates
TIR [15]	✓	✓	PF	CPU	Haskell	GP with crossover and mutation that uses a constrained representation capable of tuning non-linear parameters with OLS
TPSR [57]	✓	✓	✗	GPU	Python	Monte-Carlo Tree Search planning with non-linear optimization wrapper for generative models E2E and NeSymRes
uDSR [42]	✓	✗	✗	GPU	Python	Unification of pre-trained transformers, GP, and linear models, into a framework that decomposes the problem

Reporting the results

Performance plots: show the probability of an algorithm to achieve a certain precision level - measured as the r2 - among all independent runs.



Performance plots

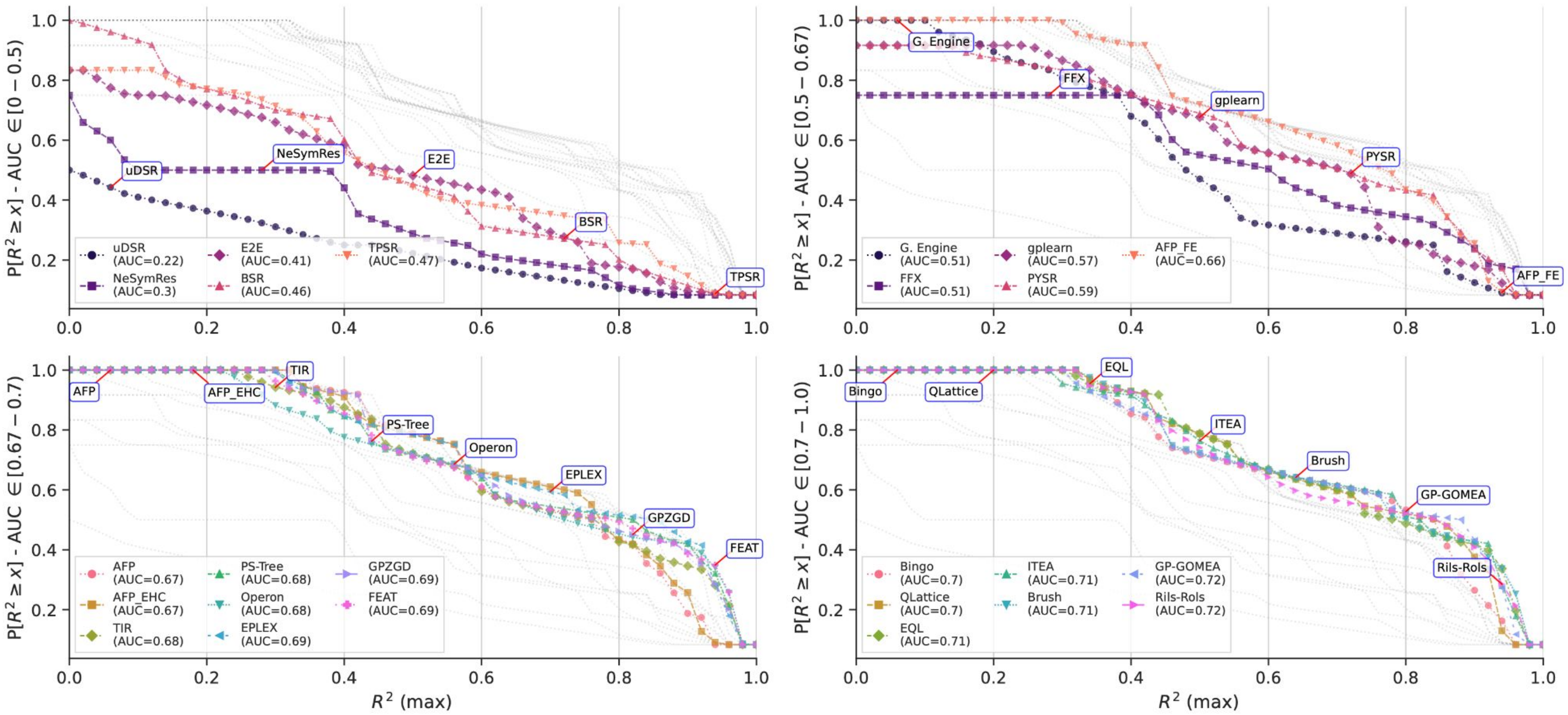


Figure 2: Performance plots for the black-box track, where the lines represent the probability of obtaining a given empirically observed R^2 value when running the experiments multiple time (i.e., max aggregation).

Trade-off

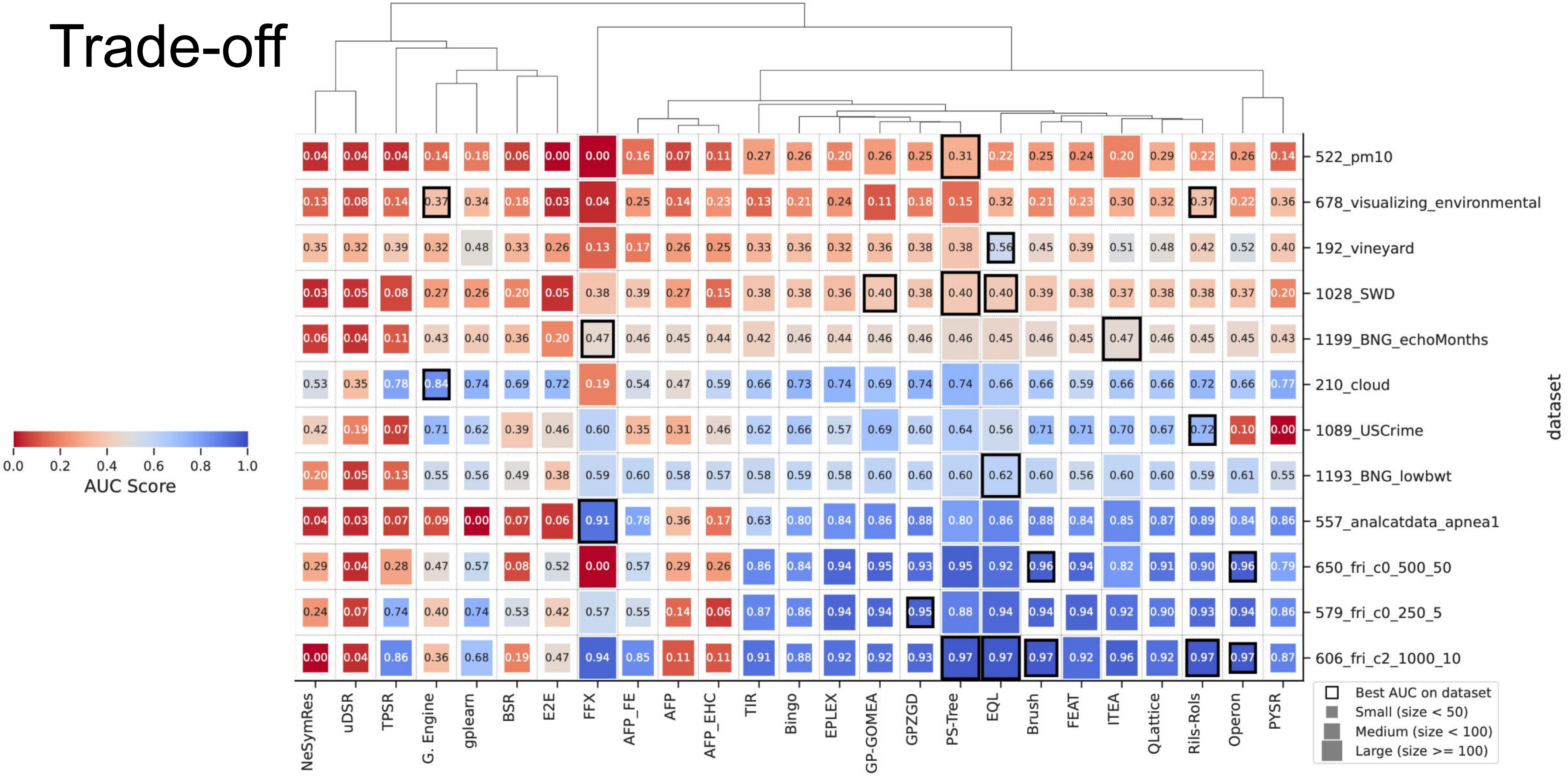
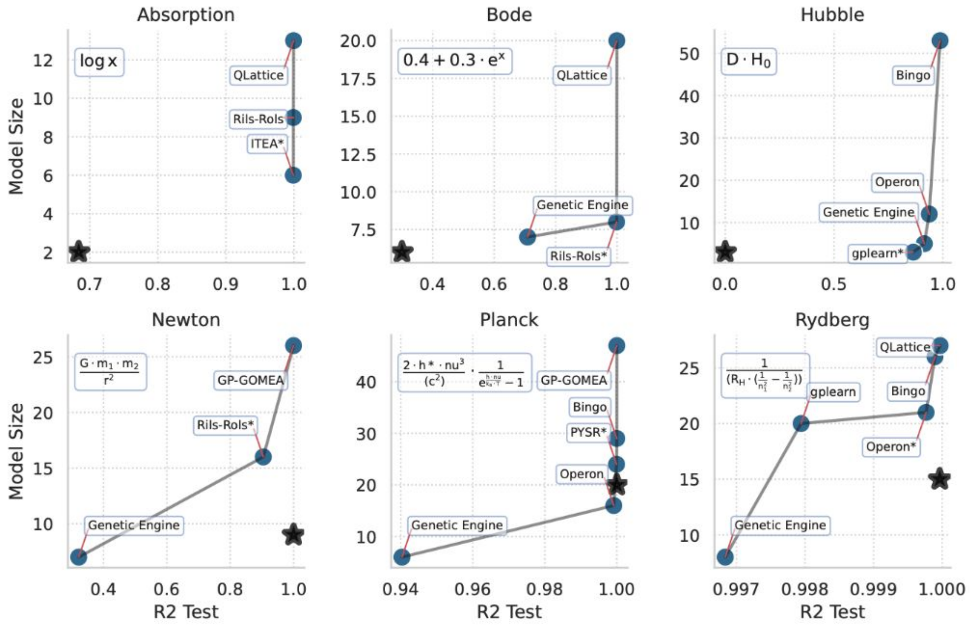


Figure 3: Cluster map of the Area Under the Curve (AUC) of Expected Performances across the 30 independent runs is segregated by algorithm and dataset. Higher values indicate better performance, while larger cells represent worse model size.

Phenomenological and first-principles

Table 3: Equation from the Pareto front closest to the governing models.

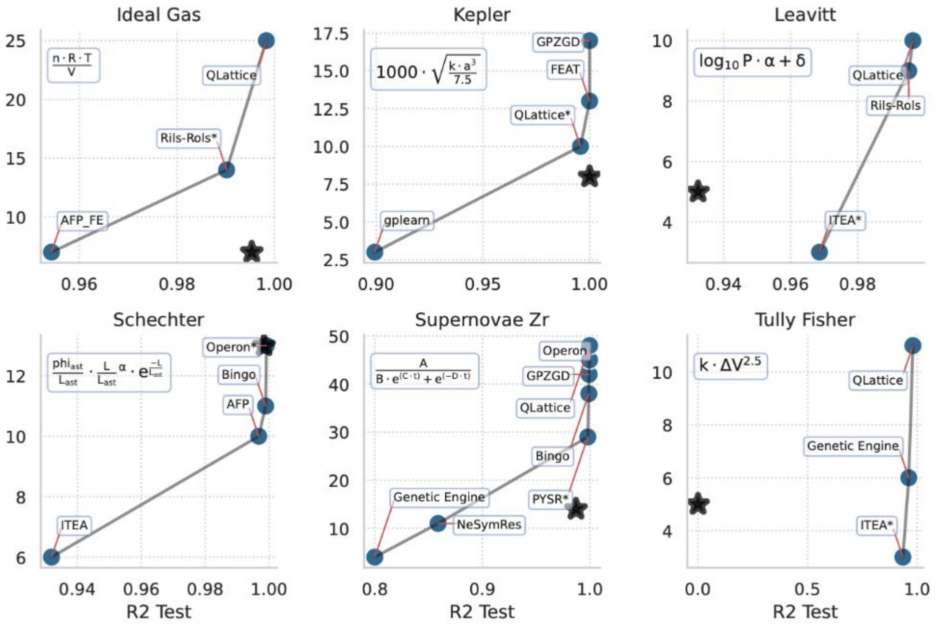
Dataset	R^2	Size	Symbolic model
absorption	1.0	6	$0.24 + 1.76 \tanh x$
bode	1.0	8	$0.34e^{1.51 \cdot n} - 0.87$
hubble	0.86	3	$0.090 + D$
ideal_gas	0.99	14	$0.69 \cdot \log n + 1.64 - 0.89 + 0.48 * e^{-V}$
kepler	1.00	10	$1.98 * \tanh(0.66 \cdot a - 0.56) + 0.78$
leavitt	0.97	3	$-0.94 \cdot \log P$
newton	0.90	16	$0.34 \cdot \frac{m_1}{0.83 \cdot m_2 - \frac{0.09}{m_1}} + 1.13$
planck	1.00	24	$0.023 \cdot \log(\sqrt{nu + 0.38} - 0.04) - 0.31 \cdot \frac{nu+0.3}{T+0.94} + 0.41$
rydberg	1.00	21	$1.01 \cdot e^{0.1 \cdot e^{1.73 \cdot n_1 - 1.31 \cdot n_2}} - e^{-0.69 \cdot n_1}$
schechter	1.00	13	$0.748 - 0.274 \cdot (\log(163.992 \cdot L + 106.737) + 1.414 \cdot L)$
supernovae_zr	1.00	29	$(\sin(0.2 \cdot x \cdot e^{-x}) - 0.04) \cdot (x + 0.72 \cdot \sin(5.46 \cdot x + 0.76) - 0.16) - \sin(x + 0.18)$
tully_fisher	0.93	3	$-0.93 \cdot \Delta V$



Phenomenological and first-principles

Table 3: Equation from the Pareto front closest to the governing models.

Dataset	R^2	Size	Symbolic model
absorption	1.0	6	$0.24 + 1.76 \tanh x$
bode	1.0	8	$0.34e^{1.51\cdot n} - 0.87$
hubble	0.86	3	$0.090 + D$
ideal_gas	0.99	14	$0.69 \cdot \log n + 1.64 - 0.89 + 0.48 * e^{-V}$
kepler	1.00	10	$1.98 * \tanh(0.66 \cdot a - 0.56) + 0.78$
leavitt	0.97	3	$-0.94 \cdot \log P$
newton	0.90	16	$0.34 \cdot \frac{m_1}{0.83 \cdot m_2 - \frac{0.09}{m_1}} + 1.13$
planck	1.00	24	$0.023 \cdot \log(\sqrt{nu + 0.38} - 0.04) - 0.31 \cdot \frac{nu+0.3}{T+0.94} + 0.41$
rydberg	1.00	21	$1.01 \cdot e^{0.1 \cdot e^{1.73\cdot n_1 - 1.31 \cdot n_2}} - e^{-0.69\cdot n_1}$
schechter	1.00	13	$0.748 - 0.274 \cdot (\log(163.992 \cdot L + 106.737) + 1.414 \cdot L)$
supernovae_zr	1.00	29	$(\sin(0.2 \cdot x \cdot e^{-x}) - 0.04) \cdot (x + 0.72 \cdot \sin(5.46 \cdot x + 0.76) - 0.16) - \sin(x + 0.18)$
tully_fisher	0.93	3	$-0.93 \cdot \Delta V$



Energy consumption

It is hard to measure energy consumption, and the libraries are not clear about their reliability over docker images.

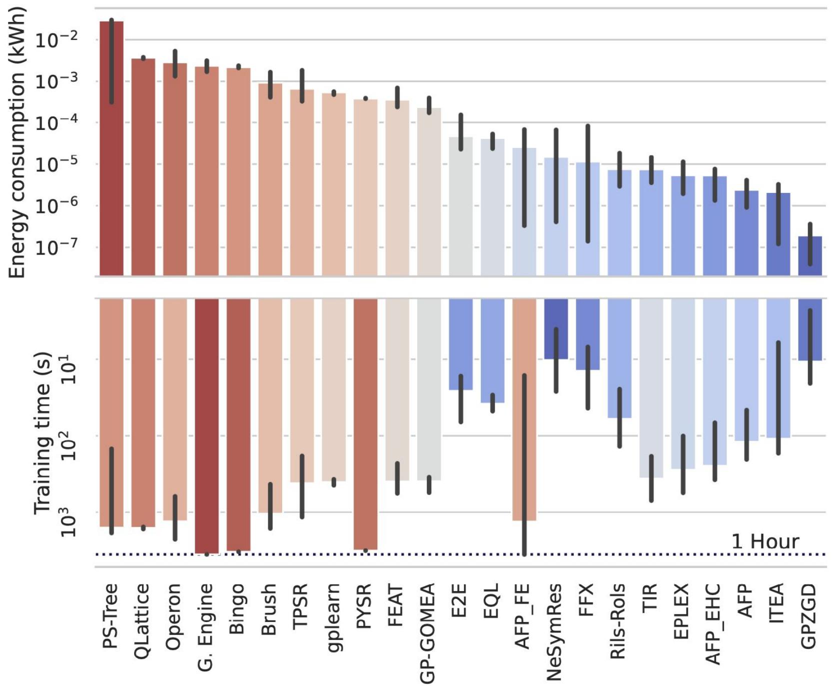


Figure 1: Median energy consumption (kWh) and training runtime for each algorithm.

Discussion

There is no *silver bullet.*

Some patterns emerged from the black-box results analysis - methods based on GP with constant optimization showed good performance, constant optimization seems essential

Phenomenological was hard for all algorithms to find the correct equation, we are still investigating. Sample size is low.

Is this the end of the road?

Not quite. Many open questions remain.

- Running the experiments is resource-intensive. What should the budget be?
- How should we compare methods across GPU and CPU?
- How can runtime be effectively managed?
- Is power consumption a fair performance metric?

Benchmarks should progress in the field and guide future directions.

Call for action

Community engagement is crucial. Code should be compatible. New results should be open. We need to discuss datasets and evaluation metrics

Visualizing results also matters—it should align with our ultimate goals. We need to define our goals and use the correct visualizations

From a practical standpoint, we need a parameter-free approach

Deprecation of methods: they are either no longer maintained or perform poorly

Final reflection

How can we ensure that benchmarks — which are getting more and more sophisticated — does not have the sole purpose of comparing algorithms with rankings, but instead actually stimulate scientific progress in SR, beyond specific datasets?

Call for action: towards the next generation of *symbolic regression benchmark*

Thank you!

Guilherme Aldeia (presenter)

guilherme.aldeia@ufabc.edu.br

Fabrício Olivetti de França (corresponding author)

folivetti@ufabc.edu.br