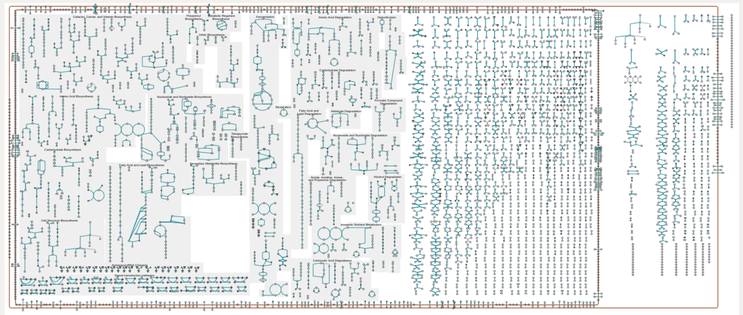
Active learning of digenic functions with boolean matrix logic programming

Lun Ai[1] Stephen H. Muggleton[2] Shi-Shun Liang[1] Geoff S. Baldwin[1]

[1]Department of Life Sciences, Imperial College London, London, UK [2]Department of Computing, Imperial College London, London, UK

Highlights

Genome-scale metabolic network models (GEMs) are blueprints for cellular engineering.



This GEM (iML1515) [3] of a widely used bacterial strain has 1515 genes and 2719 reactions.

Problem 1. GEMs do not always classify how multiple genes interact correctly.

Problem 2. Traditional logic programming cannot efficiently explore pathways in GEMs.

Problem 3. Learning interactions between multiple genes requires a large number of data.

Impact. More reliable GEMs would significantly improve the robustness of cellular engineering.

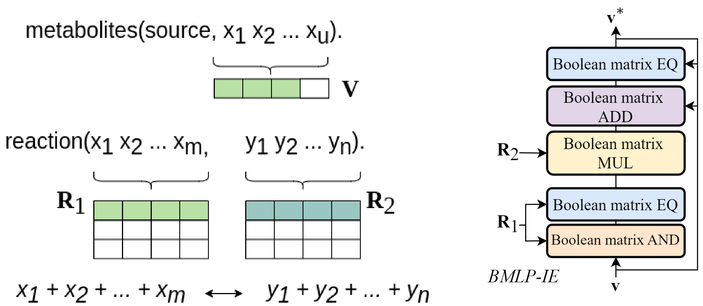
Solution. We propose Boolean Matrix Logic Programming (BMLP) to improve the runtime efficiency of datalog queries by up to 1000x. Our active learning system $BMLP_{active}$ requires 90% fewer experiments than random experimentation.

Boolean Matrix Logic Programming (BMLP)

Compared to traditional logic programming, BMLP uses boolean matrices to evaluate recursive datalog programs with arity at most two and at most two body literals, namely the H_2^2 program class [4]. Evaluating a H_2^2 program is reduced to computing the closure of boolean matrices.

$$reaction(X,Y) \leftarrow metabolites(X,Y).$$
$$pathway(X,Y) \leftarrow reaction(X,Y).$$
$$pathway(X,Y) \leftarrow reaction(X,Z), pathway(Z,Y).$$

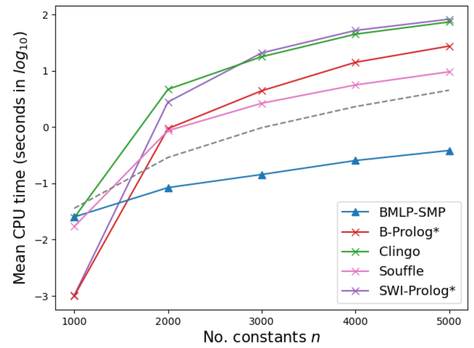
Mapping reactions to boolean matrices. We implemented a BMLP module called BMLP-IE that computes **v***, the closure of pathway products, using boolean matrix operations. Input **v** encodes source chemical metabolites. A reaction is represented in two boolean matrices, $\mathbf{R}_1$ and $\mathbf{R}_2$.



BMLP - higher runtime efficiency

Question. How much runtime improvement can BMLP modules bring?

Comparisons. BMLP-SMP [1] is an optimised module based on BMLP-IE. We compared BMLP-SMP, Souffle, B-Prolog, Clingo and SWI-Prolog in evaluating the closure of the pathway program.



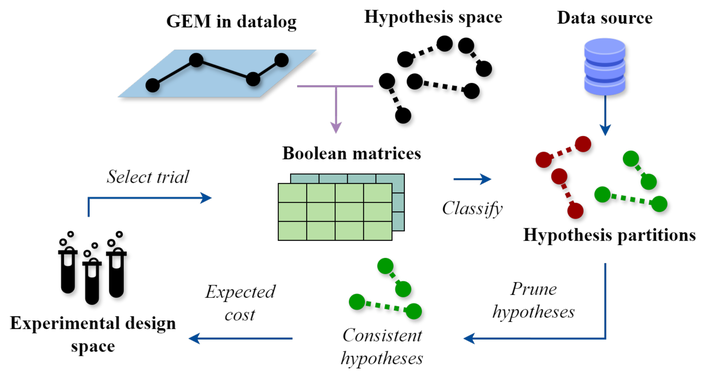
Result 1. BMLP-SMP's runtime is $O(n^3)$ and bounded by a cubically growing curve (grey).

Result 2. In the above figure [1], BMLP-SMP is at least 1 order of magnitude faster than Souffle, B-Prolog, Clingo or SWI-Prolog when the program contains ∼25,000 facts.

Result 3. In [1] we show BMLP-SMP is >1000x faster than Souffle, B-Prolog, Clingo or SWI-Prolog when the program contains ∼12,500,000 facts, since BMLP-SMP is only affected by n.

$BMLP_{active}$ - the first logic programming system to learn from GEMs

$BMLP_{active}$ selects experiments to minimise the expected value of a user-defined cost function and uses BMLP-IE to prune gene-reaction associations inconsistent with actively acquired labels.



Certain genetic mutations would block pathways, causing cells to die (positive label). $BMLP_{active}$ finds a gene-reaction association hypothesis to explain the pathway blockage and lethality. It encodes the GEM iML1515 as boolean matrices and uses BMLP-IE to classify the genetic mutation experiment labels for every hypothesis. It consults a data source to request ground truth labels. $BMLP_{active}$ iteratively refutes hypotheses inconsistent with ground truth labels.

Active learning

Posterior probability estimation. $BMLP_{active}$ uses the Minimal Description Length principle and the compression score of a hypothesis h [2] to estimate the posterior probability:

$$compression(h,E) = |E^+| - \frac{|E^+|}{pc_h}(size(h) + fp_h)$$
$$p'(h|E) = \frac{2^{compression(h,E)}}{\sum_{h_i \in H} 2^{compression(h_i,E)}}$$

E are labelled examples and E^+ are positive examples. pc_h and fp_h are the number of positive predictions and false positive coverage.

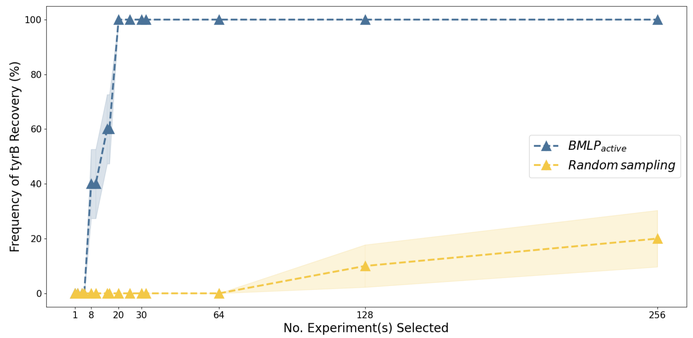
Experiment selection. $BMLP_{active}$ uses the following heuristic function from [2] to select an experiment from leftover candidate experiments T with approximated minimum expected cost:

$$EC(H,T) \approx min_{t \in T}\,[C_t + p(t)(mean_{t' \in T-\{t\}} C_{t'})J_{H_t} + (1-p(t))(mean_{t' \in T-\{t\}} C_{t'})J_{\overline{H_t}}]$$

H_t and $\overline{H_t}$ are subsets of hypotheses H consistent and inconsistent with t's label. $J_{H_t} and J_{\overline{H_t}}$ are calculated according to the entropy $-\sum_{h \in H} p'(h|E)\,log_2(p'(h|E))$ where H is replaced by H_t or $\overline{H_t}$. The probability $p'(h|E)$ is calculated from the compression function. $p(t)$ is the probability that the label of the experiment t is positive and is computed by the probability sum of consistent hypotheses $\sum_{h \in H_t} p'(h|E)$. C_t is a user-defined experiment cost function.

Questions. Can $BMLP_{active}$ re-discover a digenic function involving the key gene $tyrB$ in the aromatic amino acid biosynthesis pathways? How much experimental data does it require?

Comparisons. The random selection strategy randomly sampled N instances from the experimental design space. $BMLP_{active}$ selected N experiments from this space to actively learn from the hypothesis space. Both methods output the hypothesis with the highest compression.



Result 4. $BMLP_{active}$ successfully recovers this key digenic function with $N = 20$ experiments.

Result 5. $BMLP_{active}$ requires >90% fewer number of experiments needed compared to random experiment selection. It provides higher information gain from each experiment compared to random experiment selection.

References

[1] L. Ai and S. H. Muggleton. Boolean Matrix Logic Programming. *arXiv*, 2024.

[2] C. H. Bryant et al. Combining inductive logic programming, active learning and robotics to discover the function of genes. *Electronic Transactions in Artificial Intelligence*, pages 1–36, 2001.

[3] J. M. Monk et al. iML1515, a knowledgebase that computes escherichia coli traits. *Nature Biotechnology*, 35(10):904–908, 2017.

[4] S. H. Muggleton et al. Meta-interpretive learning of higher-order dyadic datalog: predicate invention revisited. *Machine Learning*, 100(1):49–73, 2015.

lun.ai15@imperial.ac.uk