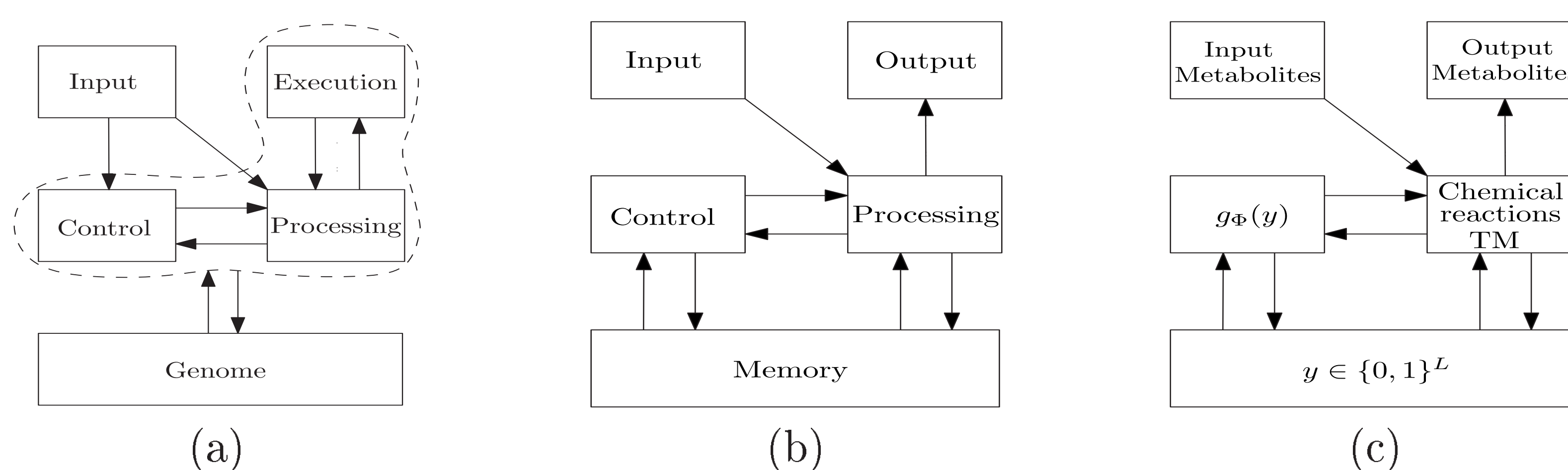


## Abstract

A protein can be thought of as a computational element, i.e. a processing unit able to transform an input into an output signal. Indeed, in a biochemical pathway, an enzyme reads the amount of reactants (substrates) and converts them into products. In this work, we consider the biochemical pathway in unicellular organisms (e.g. bacteria) as a living computer, and we program it in order to obtain desired outputs. The genome sequence is an executable code specified by a set of commands in a sort of ad-hoc low-level programming language. Each combination of genes is coded as a string of bits  $y \in \{0, 1\}^L$ . Each bit controls a gene set and therefore the chemical reaction associated with it. Through an optimal executable code stored in the “memory” of bacteria, we simultaneously maximise the concentration of two or more metabolites of interest.

## Bacteria as von Neumann architectures

Inspired by Brent and Bruck [1], who studied similarities and differences between biological systems and von Neumann computers, we propose a mapping between the von Neumann architecture and bacteria. This mapping suggests thinking of the metabolism as a Turing Machine (TM).



Comparison among biological systems (a), von Neumann architecture (b), and bacteria (c).

The bacterium takes as input the substrates required for its growth and, through its chemical reaction network, produces desired metabolites as output. The string  $y$  acts as a program stored in the RAM. Let  $Y$  be the multiset of the bits of  $y$ , and  $P(Y; p)$  be the set of all partitions of  $Y$  with  $p$  blocks. We formalise the control unit by defining the function

$$g_{\Phi} : \{0, 1\}^L \rightarrow \bigcup_{y \in \{0, 1\}^L} P(Y; p)$$

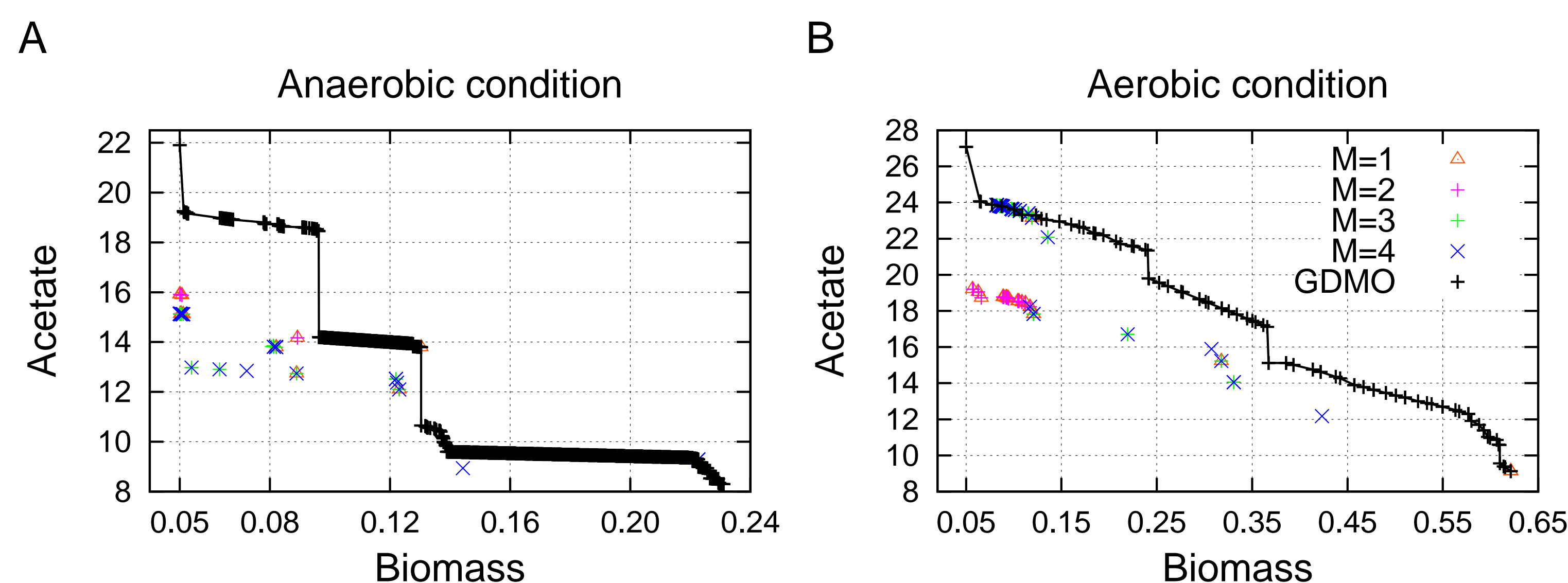
$$\bar{y} \in \{0, 1\}^L \mapsto \Pi \in P(\bar{Y}; p).$$

Each element of the partition  $\Pi$  is the submultiset  $b_s$  of all the gene sets that play a role in the reactions belonging to the  $s$ -th pathway. In other words,  $g_{\Phi}$  turns syntax into semantics.

## Genetic design of (living) computers

We program molecular machines using a novel algorithm called **Genetic Design through Multi-Objective optimisation (GDMO)**.

- Through a specific optimal code stored in the “memory” of the organism, we are able to simultaneously maximise the yield of two or more metabolites of interest.
- The genetic code, i.e. the “computation instructions” given to the machine, is represented by a Pareto-optimal string of bits  $y \in \{0, 1\}^L$ .



Maximisation in anaerobic (A) and aerobic (B) conditions with glucose uptake rate  $10 \text{ mmol h}^{-1} \text{ gDW}^{-1}$ . The Pareto fronts of GDMO are in black. The results of GDLS depend on its parameters  $M$  and  $k$  [3].

	Wild Type	OptFlux	OptGene	GDLS	GDLS	OptKnock	OptKnock	GDMO	GDMO	GDMO
Acetate	8.30	15.129 (82.3%)	15.138 (82.4%)	15.914 (91.7%)	-	-	12.565 (51.4%)	13.797 (66.20%)	19.150 (130.7%)	-
Succinate	0.077	10.007 (12877%)	9.874 (12704%)	-	9.727 (12514%)	9.069 (12362%)	-	-	-	10.610 (13659%)
Biomass	0.23	n.a.	n.a.	0.0500	0.0500	0.1181	0.1165	0.1296	0.053	0.087
k cost	n.a.	n.a.	n.a.	14	26	54	53	5	10	8

Best solutions ( $\text{mmol h}^{-1} \text{ gDW}^{-1}$ ) obtained by OptFlux ([4]), OptGene ([5]), GDLS ([3]), OptKnock ([6]) and GDMO on the *E. coli* K-12 MG1655 iAF1260 model ([7]), in anaerobic conditions with  $10 \text{ mmol h}^{-1} \text{ gDW}^{-1}$  of glucose uptake.

## Reaction networks as TM

We map the chemical reaction network to the **Minsky’s Register Machine (RM)**, i.e. a finite state machine augmented with a finite number of registers. The RM has been proven to be equivalent to the TM [2]. We define:

- The set of state species  $\{D_i\}$ , where each  $D_i$  is associated with the state  $i$  of the RM;
- The set of register species  $\{H_r\}$ , where each  $H_r$  is associated with the register  $r$  of the RM, and therefore represents the molecular count of species  $r$ ;
- The instruction  $inc(i, r, j)$  as the chemical reaction  $D_i \rightarrow D_j + H_r$ ;
- The instruction  $dec(i, r, j, k)$  as either  $D_i + H_r \rightarrow D_j$  or  $D_i \rightarrow D_k$  depending on whether  $H_r > 0$  or  $H_r = 0$  respectively.

In our FBA approach, the variables are the fluxes of the reactions in the network, therefore a high flux corresponds to both a high rate of reaction and a high mass of products.

- In the increment reaction  $inc(i, r, j)$ ,  $H_r$  is positively correlated with the reaction flux;
- In the decrement reaction  $dec(i, r, j, k)$ , when  $H_r > 0$ , it is negatively correlated with the reaction flux.

## Conclusion

Since the simulated TM can be universal, the proposed mapping between metabolism and TM allows to perform any kind of computation, through a set of species and chemical reactions characterised by their flux. In principle, this means that bacteria can carry out at least any computation performed by a computer.

A program embedded in a bacterium, whose metabolism works like a TM, could be able to implement the knockout strategy found by GDMO. The minimisation of the number of knockouts ensures a low-effort, reliable and reproducible result, allowing cells to become programmable manufacturers of biochemical products of interest.

## References

- [1] R. Brent and J. Bruck. 2020 computing: Can computers help to explain biology? *Nature*, 440(7083):416–417, 2006.
- [2] M.L. Minsky. *Computation*. Prentice-Hall, 1967.
- [3] Lun et al. Large-scale identification of genetic design strategies using local search. *Mol Syst Biol.*, 5(296).
- [4] Rocha et al. Natural computation meta-heuristics for the in silico optimization of microbial strains. *BMC Bioinformatics*, 9(1):499, 2008.
- [5] Patil et al. Evolutionary programming as a platform for in silico metabolic engineering. *BMC Bioinformatics*, 6(1):308, 2005.
- [6] Burgard et al. Optknock: a bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and Bioengineering*, 84(6):647–657, 2003.
- [7] Feist et al. A genome-scale metabolic reconstruction for escherichia coli k-12 mg1655 that accounts for 1260 orfs and thermodynamic information. *Mol Syst Biol*, 3(121):291–301, June 2007.