

Abstract

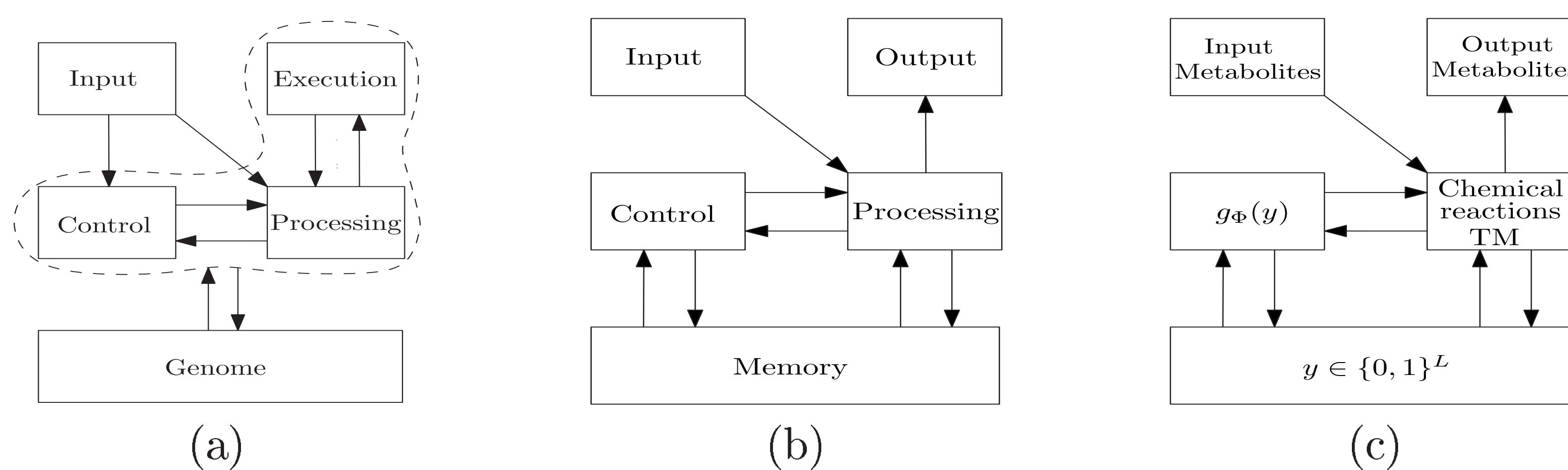
Biologically inspired computation has been recently used with mathematical models towards the design of new synthetic organisms. In a biochemical pathway, an enzyme reads the amount of reactants (substrates) and converts them into products. Therefore, in this work we consider the metabolism as a living computer, and we program it in order to obtain desired outputs.

We propose and exploit a mapping between the metabolism and a register machine (RM) to show that bacteria could have computational capability and act as **molecular Turing machines (TM)**. The reactions in the bacterium constitute the increment/decrement instructions, while the registers count the number of molecules of each metabolite. The RM is equivalent to the TM, being a multi-tape TM with the tapes restricted to work as simple counters.

We report that the density and gradient of the Pareto curve are useful tools to compare models and understand their structure, while modelling organisms as computers proves useful to carry out computation using biological machines with specific input-output conditions.

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 Π 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_{Φ} turns syntax into semantics.

Chemical reaction networks and computation

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 D of state species $\{D_i\}$, where each D_i is associated with the state i of the RM;
- The set H 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 ;
- $\varphi : D \setminus \{i_0\} \rightarrow \{(H_r, i), (H_r, j, k) \mid H_r \in H, j, k \in D\}$ a multivalued mapping.

The RM executes two basic **increment/decrement instructions** [3]:

- $inc(i, r, j)$ to increment register r by 1 and move from state i to state j according to $\varphi(i) = j$;
- $dec(i, r, j, k)$, with $H_r > 0$, to decrement register r by 1 and move from state i to state j ($\varphi(i) = j$).

Initial and halting states: $i_0, i_1 \in D$.

Register H_r : left-handed tape that stores positive integers by writing stacks of marks on the tape.

If $H_r = 0$, the machine moves from state i to state k ($\varphi(i) = k$) (**test for zero**).

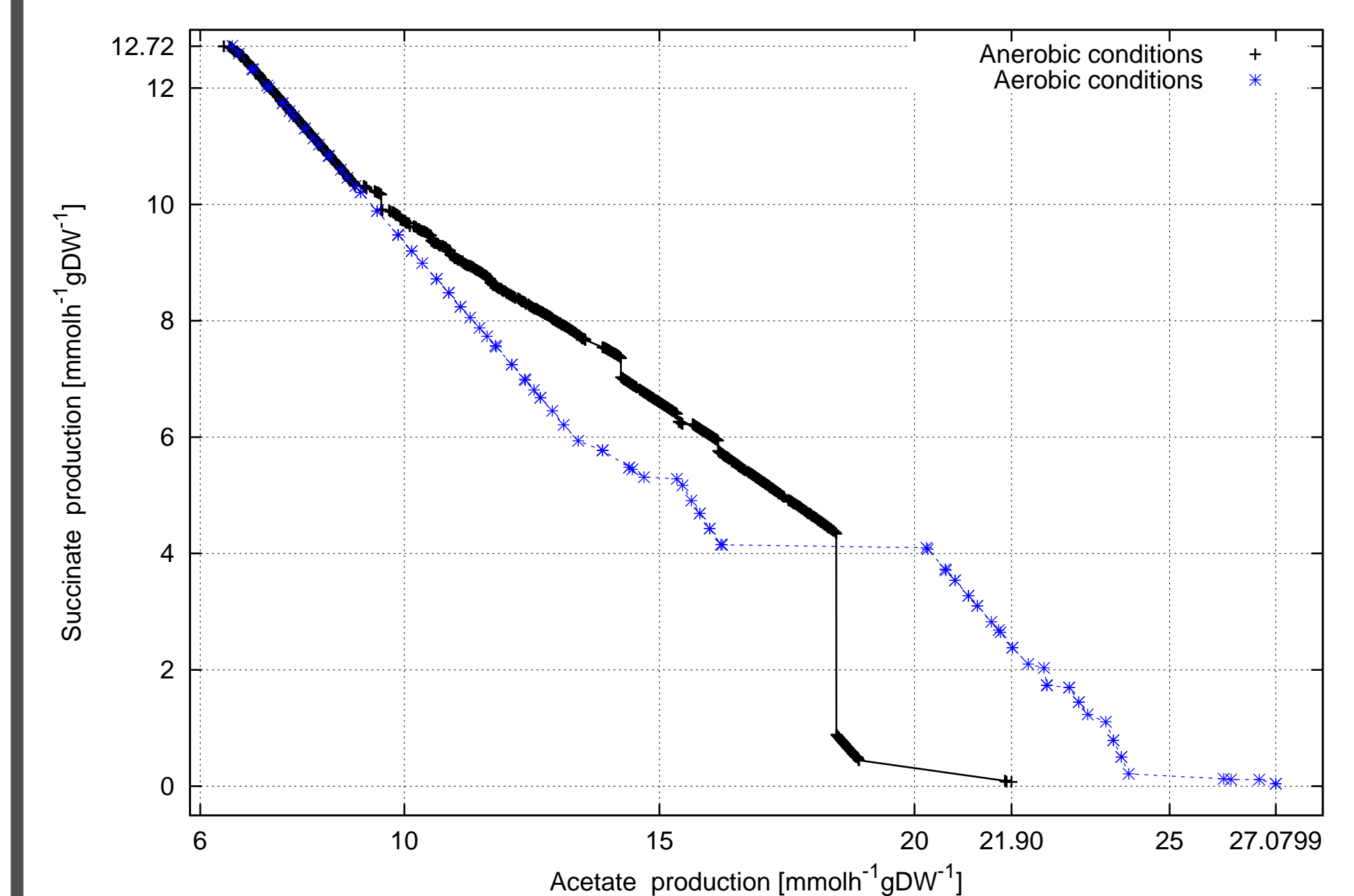
Halt instruction: halts the operation of the machine, setting the state i_1 (equivalent to the cell death, no further chemical reactions take place [4]).

- $inc(i, r, j)$ mapped to the chemical reaction $D_i \rightarrow D_j + H_r$;
- $dec(i, r, j, k)$ mapped to $D_i + H_r \rightarrow D_j$ if $H_r > 0$, or to $D_i \rightarrow D_k$ if $H_r = 0$.

Optimal molecular machines

We program molecular machines using *Genetic Design through Multi-Objective optimisation (GDMO)* [5].

- 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 [6], is represented by a Pareto-optimal string of bits $y \in \{0, 1\}^L$.



Maximisation in anaerobic and aerobic conditions with glucose uptake rate $10 \text{ mmolh}^{-1} \text{ gDW}^{-1}$.

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 [7].
- In principle, 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] D. Soloveichik, M. Cook, E. Winfree, and J. Bruck. Computation with finite stochastic chemical reaction networks. *Natural computing*, 7(4):615–633, 2008.
- [4] V. Manca. The metabolic algorithm for p systems: Principles and applications. *Theoretical Computer Science*, 404(1):142–155, 2008.
- [5] J. Costanza, G. Carapezza, C. Angione, P. Lió, and G. Nicosia. Robust design of microbial strains. *Bioinformatics*, 28(23):3097–3104, 2012.
- [6] C. Angione, P. Lió, and G. Nicosia. How to compute with metabolism in bacteria? *ERCIM News*, 91:12, 2012.
- [7] C. Angione, G. Carapezza, J. Costanza, P. Lió, and G. Nicosia. Computing with metabolic machines. In Andrei Voronkov, editor, *Turing-100*, volume 10 of *EPiC Series*, pages 1–15. EasyChair, 2012.