On the Emergence of Enzymes in an Artificial Chemistry

Giampaolo Folena[1], Germán Kruszewski [2]

[1]Centre de Recherche sur la Biodiversité et l'Environnement, CNRS-Université Paul Sabatier, Toulouse, France
[2] Computational Linguistics and Linguistic Theory Lab, Universitat Pompeu Fabra, Barcelona, Spain

Abstract

Biological systems exhibit hierarchical and intricate mechanisms that enable self-sustenance and open-ended behavior. This organizational closure is arguably one of life's hallmarks, and it is facilitated by the widespread utilization of enzymes. Enzymes enhance improbable pathways, enabling the formation of complex structures and functions. Here, we propose a model to characterize artificial enzymes within an artificial "soup" of functions. We contend that these enzymes can emerge from elementary interactions among functions, and they should foster rapid complexity growth, owing to their ability to construct auto-catalitic networks.

Introduction

Biological systems, from the microscopic intricacies of cells to the vast ecosystems of our planet, showcase a remarkable array of hierarchical and finely tuned mechanisms. These mechanisms not only ensure the sustenance of life but also facilitate its remarkable adaptability and evolutionary progress. At the core of this dynamic equilibrium lies organizational closure (Mossio and Moreno, 2015; Varela, 1979), a fundamental principle that allows biological systems to maintain their organization and functionality amidst changing environments.

Central to the achievement of organizational closure are enzymes, molecular catalysts that orchestrate and accelerate biochemical reactions with precision and efficiency. Through their nuanced regulation of metabolic pathways, enzymes enable the synthesis of essential molecules and the breakdown of complex substrates, thereby fueling the processes vital for life's sustenance and growth (Kirschning, 2021).

Enzymes accomplish these feats by navigating improbable pathways, overcoming energetic barriers, and guiding molecular transformations towards favorable outcomes. Their versatility and specificity enable the construction of intricate macromolecular structures, such as proteins and nucleic acids, essential for cellular function and genetic inheritance.

In our exploration of biological complexity, we study an artificial "soup" of functions—a computational playground

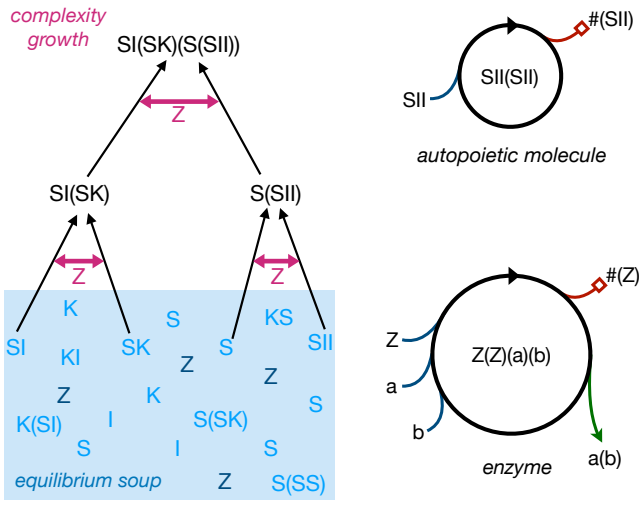

Figure 1: **(left):** Sketch of the capacity of enzymes **Z** in creating specific ligation pathways in the soup of functions (light blue sea) and therefore increasing the complexity of the system. **(right):** The metabolism of autopoietic cycles for molecules (top) and enzymes (bottom). The blue lines represent the input needed for the cycle, while the red square represent the waste of the cycle. The green arrow shows the ligation achieved by the enzyme cycle. $\#(x)$ stands for the atomic constituents of x which are released by the cycle as waste (i.e. the coding length).

where functions mimic the interplay of biochemical processes (Fontana and Buss, 1994; Hutton, 2002; Tominaga et al., 2007; Kruszewski and Mikolov, 2022). Within this artificial realm, we explore the emergence of enzyme-like entities, simple yet potent catalysts that catalyze the formation of increasingly complex molecular assemblies (Mon, 2015).

These emergent enzymes serve as the building blocks of auto-catalytic networks, self-sustaining systems where the products of one reaction catalyze subsequent reactions, leading to exponential growth in complexity (Kauffman, 1986; Hanel et al., 2005; Hordijk et al., 2011; Hordijk and Steel, 2017). Through this iterative process of self-organization and emergence, the algorithmic soup evolves, giving rise to structures and behaviors reminiscent of biological systems.

The Model: a "Soup" of Functions

We adopt the model introduced by Kruszewski and Mikolov (2022) as our algorithmic artificial chemistry as it allows for elements of unbounded complexity to emerge while also enforcing conservation laws to induce selection. It consists of a mixture of combinators (Wikipedia, 2024a), which are higher-order functions that produce new functions by selecting or recombining their arguments through function application[1]. There are atomic combinators, such as the $\mathbf{K}$ combinator, which takes two arguments xy, and returns x, or the $\mathbf{S}$ combinators, which takes three arguments xyz, and transforms them into $xz(yz)$. The chemistry follows two dynamic rules, namely, *ligation-cleavage* –two functions assemble or disassemble in proportion to their concentration–, and *reduction* –each function, with a fixed probability, acts on its arguments–. While ligation-cleavage governs the equilibrium dynamics of the mixture, reductions are the fundamental mechanism enabling the emergence of specific functions. Following Kruszewski and Mikolov (2022), we enforce the local conservation of atoms in each reduction step to prevent arbitrary function growth and enable a thermodynamic treatment of the mixture's equilibrium dynamics, as follows:

$$\alpha(\mathbf{S}xyz)\beta + z \quad \rightharpoonup \quad \alpha(xz(yz))\beta + \mathbf{S} \tag{1}$$

$$\alpha(\mathbf{K}xy)\beta \quad \rightharpoonup \quad \alpha x\beta + \mathbf{K} + y \tag{2}$$

Here, x, y, z, represent arbitrary expressions and α and β stand for some arbitrary context. The $\mathbf{S}$ combinator has been recasted into a bimolecular reaction that absorbs z from the environment, whereas $\mathbf{K}$ now releases y as a by-product.

The selection of the atomic base is critical due to two key requirements: 1) the base must support *universal computation*, 2) the base should be *efficient* in growing the complexity. Meeting the first requirement, known as Turing-completeness, is straightforward, as it has been shown that the $\mathbf{S}$ and $\mathbf{K}$ combinators are sufficient. However, this base is not necessarily efficient. To illustrate the significance of choosing an efficient base, let's consider the introduction of the $\mathbf{I}$ combinator to extend the $\mathbf{SK}$-base, which in our model acts as

$$\alpha(\mathbf{I}x)\beta \rightharpoonup \alpha x\beta + \mathbf{I} \tag{3}$$

This can be viewed as condensing the $\mathbf{SKK}$ expression of the $\mathbf{SK}$-base, reducing the coding length of the identity function from 3 to 1 (and avoiding a temporary duplication of the argument). In a system based on $\mathbf{SKI}$ (Wikipedia, 2024b) with 10^6 atoms, an autopoietic cycle emerges $\mathbf{SII}(\mathbf{SII})^* \implies \mathbf{I}(\mathbf{SII})(\mathbf{I}(\mathbf{SII})) \implies \mathbf{I}(\mathbf{SII})(\mathbf{SII}) \implies *$ which is constituted by 3 reductions (in the order $\mathbf{S}$, $\mathbf{I}$, $\mathbf{I}$). Notice that this cycle can be thought as an elementary form of metabolism, the first $\mathbf{S}$-reduction acquires an external reactant $\mathbf{SII}$, which is consumed by the cycle and returned as separated atoms (see Fig. 1:top-right). We term this cycle an "autopoietic molecule"[2] since it persists within itself and can serve as a building block for more complex functions. In the $\mathbf{SKI}$-base, this autopoietic molecule has a coding length of 6, requiring fewer combinators for its construction. However, in the $\mathbf{SK}$-base, the same molecule takes the form $\mathbf{S}(\mathbf{SKK})(\mathbf{SKK})(\mathbf{S}(\mathbf{SKK})(\mathbf{SKK}))$, increasing its coding length to 14 and the cycle length to 7. This indicates that introducing the $\mathbf{I}$ significantly simplifies the emergence of autopoietic molecules. This underscores how the efficiency of emergent structures, such as autopoietic molecules, is intricately linked to the choice of the combinator base.

Preliminary Results: How Enzymes Emerge?

Taking inspiration from nature, we investigate the emergence of enzymes within our algorithmic "soup." We define an enzyme as a function that forms an autopoietic molecule $\mathbf{Z}(\mathbf{Z})$ –where $\mathbf{Z}$ is an expression of atomic combinators– which during its cycle, combines two specific functions, a and b, into a new function, $a(b)$, releasing them into the environment as a by-product (illustrated on the right side of Fig. 1:bottom-right). Although observing its emergence in simulations poses challenges, we have managed to construct such an enzyme analytically. Currently, we are exploring methods to enhance the system to facilitate its emergence from the equilibrium soup. Similar to autopoietic molecules, achieving the emergence of enzymes hinges on selecting the right base, making it a crucial step in understanding the cost associated with the emergence and maintenance of enzyme cycles in different bases.

Perspectives: Auto-catalytic Growth

Furthermore, if enzymes can emerge and sustain their cycles, a new layer of dynamics may unfold. Beyond random ligation-cleavage events, enzyme-driven ligations (as depicted in Fig. 1:left) could open seemingly improbable pathways and construct networks of reactions reminiscent of the auto-catalytic cycles introduced by Kaufmann and colleagues (Kauffman, 1986; Hanel et al., 2005; Hordijk et al., 2011). An auto-catalytic cycle manifests when each reaction within the cycle is catalyzed by enzymes produced by the cycle itself. This self-sustenance, assuming the absence of competition, fosters a proliferation of functions participating in the cycle and increases the system's complexity growth. Notably, the discovery of auto-catalytic cycles in our model would diverge from the Kaufmann model due to (i) a non-random selection of catalyzed reactions, as they result from discovered enzymes, and (ii) the potential construction of new complex functions, which could catalyze the emergence of other selection and reproduction mechanisms.

[1] Note that (xy) stands for x operating on the argument y. Notation is left-associative, so $((xy)z)$ can be simply written as (xyz).

[2] These are recusive functions, which typically take the minimal form A(A), where A stands for an expression.

Aknowledgments

GF would like to thank Arnaud Pocheville, Marco Faggian and Matthieu Barbier for insightful discussions. GF also acknowledges support from the John Templeton Foundation (Grant No. 62220). GK acknowledges funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Grant Agreement No. 101062294.

References

(2015). *Achieving Closure in Enzymes in Artificial Chemistries*, volume ECAL 2015: the 13th European Conference on Artificial Life of *Artificial Life Conference Proceedings*.

Fontana, W. and Buss, L. W. (1994). "the arrival of the fittest": Toward a theory of biological organization. 56(1):1–64.

Hanel, R., Kauffman, S. A., and Thurner, S. (2005). Phase transition in random catalytic networks. 72(3):036117.

Hordijk, W., Kauffman, S. A., and Steel, M. (2011). Required levels of catalysis for emergence of autocatalytic sets in models of chemical reaction systems. 12(5):3085–3101.

Hordijk, W. and Steel, M. (2017). Chasing the tail: The emergence of autocatalytic networks. 152:1–10.

Hutton, T. J. (2002). Evolvable self-replicating molecules in an artificial chemistry. *Artificial life*, 8(4):341–356.

Kauffman, S. A. (1986). Autocatalytic sets of proteins. 119(1):1–24.

Kirschning, A. (2021). Coenzymes and their role in the evolution of life. *Angewandte Chemie International Edition*, 60(12):6242–6269.

Kruszewski, G. and Mikolov, T. (2022). Emergence of Self-Reproducing Metabolisms as Recursive Algorithms in an Artificial Chemistry. *Artificial Life*, 27(3–4):277–299.

Mossio, M. and Moreno, A. (2015). Biological autonomy: A philosophical and theoretical enquiry. history, philosophy, and theory of life sciences series.

Tominaga, K., Watanabe, T., Kobayashi, K., Nakamura, M., Kishi, K., and Kazuno, M. (2007). Modeling Molecular Computing Systems by an Artificial Chemistry—Its Expressive Power and Application. *Artificial Life*, 13(3):223–247.

Varela, F. J. (1979). *Principles of Biological Autonomy*. North-Holland.

Wikipedia (2024a). Combinatory logic — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Combinatory_logic`.

Wikipedia (2024b). SKI combinator calculus — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/SKI_combinator_calculus`.