

Life as a Computational System

Eduardo Bergel PhD, www.t333t.com

Life's elusive essence may lie in information. This narrative review argues that living systems are fundamentally computational or information-processing phenomena, in a specific sense: not merely by metaphor or analogy, but through concrete mechanistic processes that manipulate and preserve information. We survey advances from 2022–2025 across bioenergetics, systems biology, algorithmic information theory, and origin-of-life research to substantiate this thesis. Living organisms harness energy to stave off entropy and sustain self-organization 1, using hierarchical architectures of error-correcting, adaptive feedback loops to process information at multiple levels ² ³. We examine key evolutionary transitions—particularly eukaryogenesis—as critical inflection points where life's "information architecture" was radically restructured, enabling leaps in complexity 4 5. Formal frameworks like algorithmic information theory (AIT), assembly theory, and constructor theory provide quantitative tools to define life in terms of information and computation, yielding testable hypotheses about life's origins and universal traits. Distinguishing clearly between metaphorical invocations of computation and life's literal informationprocessing capacities, we propose a unified view: life is best understood as an emergent computational system-one that arose within the laws of physics yet exhibits novel, higher-level rules due to the processing and preservation of information. We conclude by highlighting open questions at the frontiers of this interdisciplinary synthesis and outlining how future research can further unravel life's fundamental mystery.

Introduction

What is "life"? This age-old question has gained new urgency in an era of synthetic biology and astrobiology, yet it remains notoriously difficult. Classical definitions—metabolism, reproduction, evolution, homeostasis —capture facets of life but fall short of a singular essence. A growing chorus of research suggests that **information** may be the key: living systems uniquely acquire, store, transmit, and use information to maintain their organization and function ⁶. In other words, life *processes information*. This perspective casts life as fundamentally computational, provoking both excitement and controversy. Is the cell literally a computer or is this merely a useful analogy? What distinguishes the computation occurring in living cells from the operations of a laptop or a Turing machine? Here we confront these questions directly, clarifying the senses in which life can be viewed as a computational system and where the analogy breaks down.

To ground the discussion, consider Schrödinger's 1944 insight that living organisms *avoid* the impending chaos dictated by the Second Law of Thermodynamics by feeding on "negative entropy" **1**. Modern biology reframes this: life uses a continuous flux of energy to create and preserve *information-rich* structures that resist entropy **7**. A bacterium, for example, must consume nutrients (energy) to repair DNA damage, fold proteins, and maintain gradients—all processes that uphold an internal order encoded in molecular information. Life's distinguishing feature appears to be this dynamic of information maintenance in the face of noise and decay, something no non-living system achieves to the same extent. This review explores the hypothesis that *life is, at root, an information-processing system sustained by energy flow*, and that this paradigm can unite perspectives from biochemistry to evolution and even physics.

We proceed as follows. First, we distinguish **metaphorical**, **analogical**, **and mechanistic** interpretations of "life as computation" to ensure conceptual clarity. Next, we examine how living systems implement information processing in practice: the bioenergetics that power it, the autopoietic self-maintenance that defines living organization, and the multi-layered error correction and signaling networks that give life its remarkable robustness ² ³. We then delve into evolution's major transitions through an informational lens. The origin of the eukaryotic cell (eukaryogenesis) receives special attention as an "algorithmic upgrade" in life's history ⁸, enabling complexity that prokaryotes could never attain. We discuss how this transition – and others like multicellularity – reconfigured the "computational architecture" of life.

Building on these empirical insights, we survey formal theoretical frameworks that attempt to quantify and define life's informational complexity. **Algorithmic information theory (AIT)** provides measures of complexity beyond Shannon entropy, promising to capture the *algorithmic* content of biological structures **Assembly theory**, introduced in 2021–2023, offers a complementary measure of complexity by considering an object's minimal generative steps and has already been applied to identify molecular signatures of life ¹⁰. Meanwhile, **constructor theory** (a recent formulation in fundamental physics) posits that life's unique capacities (like self-reproduction and open-ended evolution) are permitted by physical laws only if those laws allow information to be instantiated in a transferable, digital form ¹¹. These frameworks not only elucidate what life *is*, but also yield testable criteria—for instance, using assembly index to detect biosignatures on other planets, or using AIT to pinpoint when chemical systems cross into "life-like" complexity.

Throughout, we emphasize empirical grounding. Wherever possible, statements about life's computational nature are linked to specific mechanisms (e.g. DNA replication as a coded information process, or gene regulatory networks performing computational logic). We draw on a broad literature, highlighting recent discoveries (2022–2025) that push the frontier: from quantifying the thermodynamic cost of cellular information processing ¹² to mapping how genome size and organization reflect algorithmic constraints ¹³. We also address criticisms—such as concerns that viewing life as computation might be "just a metaphor"—by demonstrating the rigorous content of this view. Finally, in a concluding section, we confront open questions and outline future directions. By integrating diverse perspectives into a coherent narrative, we aim to provide a PhD-level synthesis suitable for a top-tier interdisciplinary journal, illuminating how life can indeed be understood as a computational system in a deep, non-reductive sense.

Life and Computation: Metaphor, Analogy, or Mechanism?

It is easy to call DNA a "digital code" or the brain a "biological computer," but such phrases can mean very different things. Before asserting that life is computational, it is crucial to delineate **in what sense** we intend the claim. In scientific discourse, "computation" applied to life has been used (1) metaphorically, (2) analogically, and (3) mechanistically. **Table 1** summarizes these distinctions with examples:

Sense of "Life as Computation"	Meaning and Usage	Biological Example
Metaphorical	Invokes computation as a poetic descriptor; not meant literally.	Saying "DNA is software" or calling evolution "nature's programmer" – vivid, but not a literal claim.

Sense of "Life as Computation"	Meaning and Usage	Biological Example
Analogical	Draws explicit comparisons between biological processes and human- designed computations, to gain insight.	Comparing genetic regulatory networks to electrical circuits or viewing neural firing patterns as codes. Classic cybernetics likened organisms to control systems with feedback ¹⁴ .
Mechanistic	Posits that living systems physically instantiate information-processing functions in a rigorous sense.	DNA transcription/translation as a real encoding/ decoding process; cellular signaling pathways performing decision-making logic (e.g. toggling gene expression like a switch). The genome's error- correcting features resemble those in engineered communication channels 7 3.

In this review, our focus is on the **analogical and mechanistic** levels. The metaphor "life is computation" is widespread, but unless grounded in mechanistic detail it risks being trivial. Instead, we examine how far the analogy can be pushed and where it becomes literal. For instance, the **genetic code** is often cited as evidence that life encodes information digitally much like a computer. Indeed, sequences of DNA bases (A, T, G, C) can be seen as quaternary "bits" that get translated via the ribosome into proteins. This is more than metaphor: it is a *codified mapping* akin to a lookup table, complete with redundancy and error-detection capabilities ¹⁵. Researchers in coding theory have pointed out that the DNA code has a built-in robustness; as one study notes, "the sequence of bases in DNA may be considered as digital codes which transmit genetic information," with the code's structure optimized for error resistance ¹⁵. Here the computational analogy shades into mechanism—DNA truly behaves like a stored data tape read by cellular machines (polymerases and ribosomes) that perform discrete operations (copying, editing, and executing instructions).

At the same time, **cells are not silicon computers**. No centralized CPU directs cellular operations; there is no binary instruction set separate from the chemistry. Instead, computation in life is **embedded**: it arises from the self-organizing interactions of molecules. In this sense, life's computation is **analog** and **distributed**. A single E. coli cell, for example, can "compute" optimal gene expression responses to nutrients or stress, but this happens via networks of biochemical reactions that correspond to logic circuits only in a loose sense. One influential analogy is to view metabolism and gene regulation as a network of information flow: signals (inputs) alter gene activity (processors), which produce metabolic changes (outputs) that feedback to inputs ¹⁴ ⁶. This perspective was advanced by cybernetics and systems biology, which regard organisms as engaged in continual information processing to maintain homeostasis. Such analogies have real predictive power (e.g., using control theory to model insulin regulation or ecological population dynamics), yet we must be cautious not to force-fit every aspect of life into computer metaphors.

Mechanistic computation in life appears most clearly when looking at error correction and adaptive feedback. Consider DNA replication: the DNA polymerase enzyme not only copies the genetic information but also proofreads, pausing to excise mispaired bases. This is an algorithmic if-then check akin to a computational error-correcting routine. Without it, genetic information would degrade irreversibly over generations. In the 1990s, information theorist Bernard Battail pointed out that heredity faces the same problem as any noisy communication channel: without error correction, signals (genes) would accumulate errors and eventually become gibberish **7**. He argued this had been underappreciated in biology ¹⁶.

Empirical evidence now shows multiple layers of genomic error-handling. For example, ancient, essential gene families (like HOX developmental genes) are found to be exceptionally well-conserved over evolutionary time, far more than expected by normal mutation rates. One hypothesis is that a **"nested" error-correcting code** is at work – perhaps DNA sequences and chromosomal architectures arranged in ways that preserve crucial information through a kind of redundancy or checksum mechanism ³. As a peer-reviewed study argues, the high conservation of HOX gene regions "cannot be explained unless assuming that a genomic error-correcting code resulting from a stepwise encoding exists" ³. This idea remains theoretical, but it underscores that life's informational continuity over billions of years likely relies on error correction principles familiar from computing (albeit implemented in biochemical fashion).

In summary, calling life a computational system is **not** to say organisms run on binary code or that a cell is a little laptop. Rather, it means that *information processing is a fundamental, causal factor in living systems*. Living systems instantiate physical processes that can be mapped to informational operations: they encode data (in genes, proteins, signals), perform logical operations (feedback loops, regulatory "decisions"), and even implement error-correcting codes. These are mechanistic realities, even if they do not involve electronic transistors. Throughout this review, when we speak of life's "computations" or "information processing," we refer to this concrete, mechanistic sense, while using analogies to standard computing only as guiding metaphors. With this understanding, we turn to how life's computational nature manifests, starting with the energetic and organizational foundations that make such information processing possible.

Bioenergetics and Informational Complexity

Living systems are consummate energy transducers. They must be, in order to create and maintain improbable states of order (low entropy) rich in information. Schrödinger's insight that life feeds on negative entropy has evolved into a modern science of **biological energetics**: how organisms extract usable work from their environment and dissipate entropy back out, all while preserving internal structure. Crucially, this energetic expenditure is tied to information processing. It takes work to copy DNA, to correct errors, to power molecular motors that shuttle signals. In this section, we explore the intimate link between energy and information in living systems, and how constraints in energy availability can impose limits on biological complexity.

Claude Shannon's information theory famously established that *information is physical*: to send, copy, or erase a bit of information requires physical resources (energy, channel capacity, etc.). Landauer's principle quantifies this, showing a minimum energy cost for erasing a bit at a given temperature. Living cells operate in regimes where these costs are non-negligible. For example, the *proofreading* step in DNA replication consumes extra ATP energy; the benefit is reduced error rates (information fidelity). Organisms have evolved remarkably efficient ways to approach these physical limits. Recent work in **stochastic thermodynamics** has begun to analyze biological processes in this light, treating cells as information engines. A 2025 review by Cao & Liang surveys how nonequilibrium thermodynamics can elucidate the fundamental limits of biological functions, emphasizing examples like molecular machines and *error correction mechanisms* in biology ¹². By applying these principles, researchers are quantifying trade-offs such as the energy cost of maintaining a high-fidelity genome or of sensing the environment accurately. One outcome of this approach is the realization that biological networks often operate near optimal points balancing energy expenditure and information gain – essentially, organisms spend just enough energy to gather and use information that improves their survival, but not much more, reflecting an evolutionary tuning.

The "Power" Behind Complexity: Mitochondria and Eukaryotic Innovation

A striking illustration of how energy availability ties to informational complexity is seen in the evolution of eukaryotes. All complex multicellular life (plants, animals, fungi) descends from a singular ancestral event: the merger of two simpler cells (an archaeal host and a bacterial endosymbiont) to form the first eukaryotic cell. This endosymbiosis gave rise to mitochondria – the cell's power plants – and coincided with an explosion in genome size and regulatory complexity. Why? One long-standing hypothesis is that mitochondria provided a huge boost in energy per gene, lifting a previous bioenergetic constraint that had capped prokaryotic complexity ¹⁷ ¹⁸. Nick Lane and colleagues argued that a typical eukaryotic cell enjoys orders of magnitude more energy *per gene* than a bacterium, because the mitochondria's internal membranes massively expand the energy-generating surface area relative to volume ¹⁹ ²⁰. In Lane's view, this **"extreme genomic asymmetry"** – tiny bioenergetic genomes inside mitochondria supporting a large nuclear genome – allowed eukaryotes to afford many more genes (including lots of non-coding DNA) and to express them at far higher levels ¹⁷. In principle, this freed eukaryotic lineages to explore vastly larger combinatorial spaces of gene regulation and cell specialization, paving the way for innovations like multicellularity and complex developmental programs ²⁰.

Interestingly, this bioenergetic theory has not gone unchallenged. Comparative cellular physiology by Lynch and Marinov (2015-2017) suggested that on a per-cell-volume basis, eukaryotes do not produce significantly more energy than bacteria (21). In their data, mitochondria did not greatly enhance total ATP per unit volume, calling into question the magnitude of the energy advantage. Lane and others countered that energy per gene (rather than per volume) is the relevant metric, since eukaryotes concentrate power in mitochondria while outsourcing most genetic functions to the nucleus ¹⁹. The debate continues, but a synthesis is emerging: the key may not be raw energy output, but how energy is organized and coupled to information. In eukaryotes, having separate compartments (nucleus vs. mitochondria) fundamentally changed how genomes are regulated and protected. The mitochondrial genome shed most of its genes to the nucleus, enabling streamlined energy production, while the nucleus expanded and evolved rich internal regulation (splicing, chromatin, etc.) ²² ¹⁷. In essence, eukaryotes reorganized the flow of energy and information: mitochondria handle most ATP generation but under tight nuclear control, and the nucleus, relieved from energy-production duty, could evolve new layers of gene regulation complexity. This is a prime example of **energy-information coupling** in evolution: a new energy mechanism (endosymbiosis) didn't just "add power" like a bigger engine, it transformed the cell's computational architecture.

One concrete consequence of this transformation is the advent of **a dedicated information-processing organelle** – the nucleus – separated from the bioenergetic organelles. This has an analogy in computer architecture: by separating "memory storage" (nucleus housing DNA) from "power supply and processing" (mitochondria fueling metabolism), the eukaryotic cell achieved a scalable architecture ²³ ²⁴. Prokaryotes, in contrast, have everything in one compartment, which imposes trade-offs: DNA is located in the cytoplasm where metabolism occurs, so genome expansion slows processes or becomes energetically costly. Eukaryotes solved this by putting DNA in a vault (nucleus) and managing metabolic reactions in mitochondria and cytosol, with regulated exchange between them. This compartmentalization is what allowed eukaryotic genomes to expand massively (billions of base pairs, with introns and regulatory elements) without proportionally crippling the cell's efficiency ²⁵ ²⁴. The presence of **introns** (non-coding segments spliced out of mRNAs) is particularly noteworthy: introns are an energy cost (the cell must transcribe extra DNA and then splice it out), but they enable **modular information processing** – exons can be shuffled or alternative-spliced to create multiple proteins from one gene, and splicing provides a

regulatory checkpoint. This looks like inefficient redundancy until one appreciates it as an algorithmic innovation with long-term payoffs in evolvability.

Thus, eukaryogenesis highlights how increasing energy availability can synergize with information architecture to unlock new levels of complexity. It's not simply "more energy = more complexity," but rather that *energy must be harnessed in an informationally structured way*. Mitochondria were only transformative because their integration rearranged the cell's information flow (gene-metabolism coupling) ²⁶ ²⁷. This theme recurs throughout life's history: major increases in complexity often coincide with innovations in how energy is utilized to process information. We will revisit eukaryogenesis in more detail later as a case study of a key transition. First, we examine the general principles of how living systems self-organize and manage information, which sets the stage for understanding those evolutionary leaps.

Information and Autopoiesis: The Self-Maintaining Computational Network

A living cell is not a passive collection of molecules; it is an active, self-maintaining network of chemical reactions. Maturana and Varela coined the term **autopoiesis** (from Greek "auto" = self, "poiesis" = creation) to describe this property: an autopoietic system continuously regenerates and maintains its own components and boundaries, separating itself from the environment while exchanging matter and energy to sustain order. In simpler terms, life is a *self-producing* system. This concept, introduced in the 1970s, was an attempt to capture the organizational closure of living cells – the way every part of the cell is produced by interactions of other parts, forming a holistic, self-perpetuating network.

Today, autopoiesis is recognized as a key feature that any definition of life must reckon with. But how does it relate to information and computation? Biophysicist Howard Pattee and others have argued that the secret of life's autonomy lies in a duality: living systems carry a **semantically rich information structure** (e.g., DNA, or more abstractly, the "knowledge" of how to build and sustain the organism) and a **dynamic machinery** that executes processes according to that information. Autopoiesis marries these two: the cell's machinery (enzymes, membranes, etc.) is continually rebuilding itself according to instructions and signals (information) that are themselves maintained by that machinery. It's a bootstrap: hardware and software coevolving. One researcher, William Hall, emphasized that "autopoiesis and the construction of knowledge are inseparable aspects of physical phenomena scalable to many levels of organization" (2). In other words, at the core of an autopoietic system's self-production is an informational loop—what the cell "knows" (in terms of structure and state) directly contributes to what it does, and vice versa, across scales from molecular to ecological (2).



Figure 1: A dividing cell (human HeLa cells in anaphase, with DNA in blue and structural filaments in green) exemplifies autopoiesis – the self-produced, self-maintained nature of life. Each daughter cell inherits a copy of the genetic information (blue) and a share of the cellular machinery, allowing the autopoietic process to continue. Information (genes) guides the assembly of structure (proteins, membranes, shown in green and orange), which in turn ensures the faithful copying and segregation of that information.

Autopoiesis highlights an important point often lost in the computational metaphor: **computation in living systems is not about abstract symbol manipulation divorced from material reality; it is intrinsically bound up with constructing and preserving a physical entity.** A cell computes *itself into existence* moment by moment. This sets life's computation apart from, say, a computer program computing an answer and then terminating; a cell's "output" is ultimately its own continued survival and reproduction. The concept of autopoiesis has been scaled up to explain larger systems (organisms, ecosystems, even societies) and links to ideas in cognitive science (e.g. enactive cognition suggests that an organism's perception and actions form a self-sustaining loop of information). For our purposes, the takeaway is that any computational view of life must account for this circular, self-referential information processing. The "program" and the "machine" in biology are deeply intertwined.

Seen through the autopoietic lens, many classical hallmarks of life – metabolism, homeostasis, adaptation – are manifestations of a system maintaining its own integrity by informational feedback. For example, homeostasis (keeping internal conditions within bounds) can be seen as a real-time computation where sensors detect deviations and effectors correct them, implementing a control algorithm. On the molecular level, consider the heat-shock response: when proteins start to misfold at high temperature, cells produce chaperone proteins that help refold or degrade the misfolded proteins. This is a feedback control system: the cell "senses" the informational state of its proteome (via proteins that detect misfolded shapes) and triggers a compensatory program encoded in DNA (the heat shock genes). It's as if the cell runs a subroutine to restore order – a clear example of a built-in algorithm honed by evolution to ensure the autopoietic network doesn't fall apart under stress.

Autopoiesis also dovetails with the idea of emergence in complex systems. A living cell exhibits properties (like reproduction or purposeful behavior) that none of its constituent molecules have in isolation. These emergent properties arise from the network of interactions. As one textbook example puts it: "Life is an emergent property; none of the component molecules of a cell are alive, only a whole cell lives". Emergence is essentially collective computation: numerous components interacting generate higher-level order. We see layered emergent phenomena across biology's hierarchy: molecules form organelles, organelles form cells, cells form tissues, and so on, each level bringing new information processing capabilities (neurons collectively produce consciousness, colonies of insects collectively make decisions, etc.). Crucially, each level also introduces constraints and error-correcting features that stabilize the emergent order. The genome's stability is maintained by molecular repair systems; a tissue's stability might be maintained by cell-cell signaling that regulates growth (avoiding runaway cancer, for instance). This perspective aligns with autopoiesis by emphasizing how nested loops of information and causation make life robust yet adaptable. At each level, new "rules" emerge that cannot be predicted by full knowledge of the lower level 28 29, because the collective dynamics generate novel informational structures (like a neural network's memory or an immune system's recognition repertoire). Life's computational nature is thus inherently hierarchical and multi-scale. We find information encoded in DNA sequences, in metabolic network states, in neural circuits, in group behaviors, all at once – each level feeding into and constraining the others ¹⁴. This nesting is reminiscent of modular programming or layered architecture in computer science, but achieved organically through evolution.

In summary, autopoiesis provides a foundational view of life as a self-sustaining information processing loop. It reminds us that the "computations" of life are always geared toward one overarching task: *persisting as an organized system*. Unlike a man-made computer, which dutifully runs any algorithm we load into it, a living system's computations are inextricably self-referential – they are about maintaining the very machine doing the computation. This principle will be important as we consider evolution next: evolution, in a sense, is the long-term computation by which life explores possible self-maintaining configurations, with natural selection favoring those that compute themselves most successfully. The next section examines how evolutionary transitions can be viewed through changes in life's information processing architecture.

Evolutionary Transitions as Information Upgrades

Life's history on Earth can be viewed as a series of major transitions, each of which introduced a new way of storing, transmitting, or processing information. The framework of **"Major Evolutionary Transitions"**, originally articulated by John Maynard Smith and Eörs Szathmáry, includes events like the origin of replicating molecules, the emergence of chromosomes (groups of genes), the advent of the genetic code, the rise of eukaryotic cells, the development of sexual reproduction, multicellularity, and the emergence of societies and language. Strikingly, every transition on that list corresponds to an innovation in information handling: for example, chromosomes and sexual reproduction enabled more efficient genetic information mixing; multicellularity created new channels for cellular communication and specialization; language externalized information into culture. Here, we will focus on a few critical transitions to illustrate how viewing them through the lens of computation and information yields new insights. In particular, we highlight **eukaryogenesis** (the origin of the eukaryotic cell) as a watershed event, and also touch on the emergence of multicellular life and the origin of life itself, all as case studies in life's evolving "information architecture."

Breaking the Prokaryotic Ceiling: The Algorithmic Phase Transition of Eukaryogenesis

As introduced earlier, prokaryotes (bacteria and archaea) remained relatively simple for billions of years – they are extraordinarily successful at small sizes, but even the most complex prokaryotes (like photosynthetic cyanobacteria) don't approach the structural or genomic complexity of eukaryotes. This suggests there were constraints holding back complexity. We've discussed the bioenergetic constraint (surface-area-to-volume limits on energy production) and how the mitochondrial endosymbiosis alleviated that. However, recent research has illuminated a second, subtler constraint: a **computational (or algorithmic) constraint** on exploring genomic complexity. In 2025, Muro *et al.* published an analysis of protein and gene length distributions across ~33,000 genomes from all domains of life ³⁰ ³¹. They found that throughout prokaryotic evolution, average protein-coding gene length slowly increased, following a roughly log-normal distribution that can be modeled by a multiplicative stochastic process (essentially, random incremental changes) ³¹ ³². However, when gene lengths reached a certain threshold (around **1500 nucleotides on average**), a dramatic change occurred ¹³. This corresponds to the time of eukaryogenesis, roughly 1.5–2 billion years ago. Beyond this point, gene lengths, protein lengths, and genome sizes no longer followed the same trends – it was as if life's "genetic algorithm" underwent a phase shift.

Muro et al. characterized this as an "evolutionary algorithmic phase transition." In the prokaryotic phase, increasing gene length was "computationally simple": as long as proteins were short, exploring sequence space by random mutation and selection was feasible 33 34. But as genes got longer, the space of possible sequences grows exponentially, making it algorithmically infeasible for blind evolution to find new functional proteins by random search 8. They postulate that by the time average genes ~1500 bp, prokaryotes were hitting this wall - a combinatorial explosion of possibilities that evolution couldn't efficiently navigate with existing mechanisms. The solution, which arose in the lineage leading to eukaryotes, was the incorporation of non-coding sequences (introns) and the advent of a nucleus with splicing. In essence, eukaryotes introduced a new algorithm: splicing and modular genes significantly reduce the complexity of searching for new proteins ⁸. Instead of one continuous coding sequence evolving stepwise, intron-exon structures allow shuffling and recombination of functional modules (exons) and alternative splicing gives multiple protein outputs from one gene. This "decoupling" of gene and protein length circumvented the previous linear constraint. As Bascompte (one of the authors) explained, in early life "increasing the length of proteins and their corresponding genes was computationally simple" but eventually "the search for longer proteins became unfeasible" - the tension was resolved "abruptly with the incorporation of non-coding sequences... With this innovation, the algorithm for searching for new proteins rapidly reduced its computational complexity" 8. The nucleus and spliceosome made the process nonlinear, breaking the algorithmic stalemate.

From this perspective, eukaryogenesis was an **information-processing revolution** as much as an energetic one. The eukaryotic cell represents a distinct phase of life where a new strategy for managing genetic information unlocked further complexity. Supporting this, the study found signatures of critical phenomena around that transition (like critical slowing down, which in physics indicates a system poised at a phase change) ³⁴ ³⁵. After eukaryotes emerged, the path was opened for other major transitions. Indeed, Muro *et al.* remark that once eukaryotic complexity took hold, it "unlocked the path toward other major transitions – such as multicellularity, sexuality, and sociability" ⁵. The eukaryotic genome, with its vastly expanded non-coding regions and the capacity for regulatory innovation, provided the substrate for those future evolutionary experiments.

It is important to note that eukaryogenesis appears to have been a **singular**, **extremely improbable event** – perhaps the hardest step in life's history, as evidenced by its unique occurrence (all eukaryotes share a common ancestor). Some astrobiologists even equate it with the Great Filter (a step so unlikely that it might explain the Fermi Paradox of why we haven't detected other complex life) ³⁶ ³⁷. The computational perspective adds a layer to this: the improbability might lie in the necessity of a chance merger of two cells that solved a deep algorithmic/energetic constraint in one stroke. The archaeal host provided a permissive environment for genome expansion (with budding internal membrane systems perhaps), while the bacterial symbiont not only gave energy but forced the evolution of a nucleus and introns due to the relocation of genes and the need to manage the combined cell ⁴ ¹⁹. Such a perfect storm is unlikely to repeat often. Once it did happen, however, natural selection could take over to refine and elaborate the new computational architecture.

Multicellularity and Beyond: Hierarchical Information Integration

Following eukaryogenesis, another major transition was the rise of multicellular organisms. Multicellularity evolved multiple times independently (in animals, plants, fungi, algae, etc.), suggesting it might be a more accessible innovation given the eukaryotic toolkit. From an information standpoint, multicellularity represents a shift where information processing is distributed across a collective of cells rather than confined to one. Individual cells specialize and communicate; thus multicellular life required the invention of **cell-cell signaling and gene regulatory networks** that operate on a tissue/organism-wide scale. Developmental biology is essentially the study of how one cell (the zygote) gives rise to an organized body through a cascade of information exchanges: chemical gradients, signaling molecules, gene expression cascades – in effect, a *computational process* orchestrating millions of cells. The instructions for this process are encoded in the genome but are executed through cell interactions.

An illustrative concept is **positional information**, introduced by Lewis Wolpert: cells in a developing embryo figure out their position (say, along the head-to-tail axis) by reading concentrations of morphogen molecules, and then execute genetic programs accordingly (e.g. "if morphogen A level is high and B is low, become head tissue"). This is akin to a distributed computation where each cell performs a simple logical function based on inputs, but the global result is the formation of a complex pattern (the body plan). Importantly, multicellular organisms evolved mechanisms for error correction and robustness in development – for example, many embryos can regulate or compensate if some cells are removed or if gene dosage changes, adjusting signaling to still form a proper body plan. Such **fault-tolerance** is reminiscent of robust computation systems that can handle errors. The Hox gene network (which patterns body segments) is a classical example: it's a highly conserved, hierarchically organized gene circuit that ensures the right structures form in the right places. The conservation and redundancy in developmental gene networks (backup genes, overlapping signals) indicate that evolution built in error-correcting and fail-safe features to these multicellular computations – embryos that could "catch mistakes" in the developmental algorithm had a better chance to survive.

From a broader view, the transition to multicellularity created a new **level of information hierarchy**. Now natural selection could act not only on genes and individual cells, but on networks of cells and emergent traits (like organ function or behavior). The genome had to evolve new ways to encode not just proteins, but programs for assembling diverse cell types into a coherent whole. One could say the genome's role shifted from just specifying a cell's components to also specifying an *algorithm for development*. This is evident in phenomena like regulatory DNA sequences that act as logic gates controlling gene expression in precise spatial patterns. For instance, an enhancer sequence might integrate inputs from multiple transcription

factors (each maybe indicating presence of a certain signal or cell lineage) and drive a gene's expression only if a particular combination of conditions is met – essentially implementing a Boolean logic rule (e.g., "gene on if A AND B are present, but not C"). Such logic ensures the gene is active only in a certain tissue. These are hard-wired computations evolved over time, and we can even manipulate them now with synthetic biology, treating them as programming elements.

Going one step further, consider the emergence of **nervous systems** in animals – another threshold in life's informational complexity. Neurons are cells specialized purely for rapid information processing and communication. With neurons and brains, evolution enabled real-time computational modeling of the environment. A predator can integrate sensory data, predict prey movements, and coordinate a chase; a primate can remember fruit locations and plan social strategies. The brain's evolution is beyond our scope to discuss in detail, but it's clearly an organ of computation, converting experiences (inputs) into behavior (outputs) to enhance survival. It operates on electrical and chemical signals which obey the same principles of information theory (signals must be above noise, error-correction in neural codes can be found, etc.). The cost is high – human brains consume ~20% of our basal energy on just 2% of our body mass – again underscoring the energy-information link. Natural selection has deemed that energy worthwhile for the information advantage it provides.

Finally, the emergence of **culture and technology in humans** can be seen as life's information processing extending beyond biology into external media. With humans, genetic evolution built a brain capable of language, and language enabled information (knowledge, norms, skills) to be stored outside of genomes – in brains, and eventually in written records and digital computers. This is sometimes called the transition to a *techno-cultural* evolution, which is orders of magnitude faster than genetic evolution. It represents life's computational system offloading some of its processing to inventions like writing and computing. While this is far afield from classical biology, it's arguably a continuum: the same drive to preserve and process information that characterizes life now manifests in how human society records history or designs algorithms. We won't explore this realm deeply, but it's a reminder that *if* we consider life fundamentally as a computational process, then our current age of AI and biotechnology is just a new domain of that process – life's self-computation has bootstrapped itself into new substrates.

The Origin of Life: From Chemistry to Information

No discussion of life's fundamental nature would be complete without addressing the ultimate transition: the origin of life from non-life. Here, we are in the realm of hypothesis and experiment without certainty, but the consensus is that life began when a set of chemical reactions became organized in a self-sustaining, self-reproducing network – in short, when autopoiesis and heritable information storage first emerged. In computational terms, this was when *inanimate chemistry began to actively process information*, specifically when it began to use records of the past (initially perhaps molecular templates) to guide future behavior (the production of more of those templates).

One influential idea is that life's origin was marked by a **phase transition in information flow** – a point where the system stopped being dominated by mere thermodynamics and started being *information-driven*. Walker and Davies (2013) have described the origin of life as the transition where informational constraints (like a genetic code) took over from purely chemical constraints, enabling open-ended evolution. More concretely, it's the point where chemical reaction networks began storing information that affected their kinetics (for example, a replicating polymer carries the information that influences the rate of production of the same polymer). This is essentially the birth of *algorithmic causation* in chemistry: outputs (like a template

strand) feed back as inputs (catalyst or template for the next generation). There is ongoing work to formalize this. Some researchers talk about **"assembly information"** or "casual replication" to identify when a chemical system becomes life-like. As we will discuss in the next section, **assembly theory** attempts to quantify at what point a collection of molecules shows evidence of selection (and thus hidden information) by measuring their complexity. An exciting application is in astrobiology: by measuring molecules in extraterrestrial samples, one might detect those with complexity that is overwhelmingly unlikely to arise by random chance – a potential biosignature of life's presence ¹⁰ ³⁸.

Another concept relevant to the origin is the **error threshold** originally described by Manfred Eigen. Early replicators (likely RNA molecules or their precursors) had to be short because replication errors would accumulate in longer molecules (this is analogous to the constraint on gene length we discussed earlier). If a replicator is too long and copying is too error-prone, information cannot be preserved – the system falls into an "error catastrophe." Calculations show there's a maximum genome length for a given error rate beyond which hereditary information is lost each generation. This implies that the first life likely had to evolve error-correction mechanisms or higher fidelity replication to break through to larger genomes. A minor improvement in fidelity could have had outsized effects: one study noted that a relatively small increase in replication accuracy could allow genomes to grow enough to encode more functions, kicking off a virtuous cycle toward complexity ³⁹ ⁴⁰. This again is an example of a computational threshold (information fidelity) gating an evolutionary transition. Crossing that threshold – perhaps via the evolution of a ribozyme that proofreads, or the advent of double-stranded DNA which is more stable – would constitute an "algorithmic upgrade" in early life, enabling it to preserve larger information sets and thus evolve new capabilities.

Though the origin of life is still not fully understood, current research strongly emphasizes **information**: how did a molecular system begin to store instructions, how did those instructions become self-referential (templates encoding the machinery that reads the templates), and how was noise in this system controlled? The leading scenarios (RNA world, metabolism-first, peptide-RNA coevolution, etc.) each grapple with those questions. The computational perspective doesn't solve origin-of-life, but it provides language to formulate hypotheses: e.g., "life began when a chemical network obtained a primitive memory (like autocatalytic templates) and a rudimentary code that guided assembly processes." This is a hypothesis that can be explored by experiments – for instance, synthetic biology efforts to create protocells aim to see if we can assemble a system that spontaneously starts using information (like a genetic polymer) to direct its metabolism and reproduction. If achieved, that would be the ultimate proof of concept that life is a computational system: we would have instantiated the algorithm of life artificially.

Quantifying Life's Information: AIT, Assembly Theory, and Beyond

Thus far, we have described life's computational nature qualitatively. To elevate this discussion to a rigorous, PhD-level analysis, we need quantifiable definitions and measures. How can we objectively measure the "information" or "complexity" in a living system? What formal frameworks allow us to test whether a given system is alive or life-like based on its information processing? In this section, we highlight three interrelated approaches that have gained prominence: **Algorithmic Information Theory (AIT)** applied to biology, **Assembly Theory (AT)**, and **Constructor Theory's** perspective on life. Each provides a lens for quantification: - AIT deals with Kolmogorov complexity and algorithmic randomness, giving a way to quantify how complex a biological pattern is in terms of the shortest description or program that produces it. - Assembly Theory introduces the concept of assembly index, focusing on how many steps of combination are needed to build an object, thus highlighting evidence of selection or evolution.

Constructor Theory (especially the "Constructor Theory of Life") attempts to characterize life in terms of possible vs. impossible transformations in physics, boiling down life's requirements to the presence of certain information media and tasks.

Algorithmic Complexity in Biology

Algorithmic Information Theory, pioneered by Solomonoff, Kolmogorov, and Chaitin, marries Shannon's information theory with Turing's computability theory ⁹. In plain terms, the Kolmogorov complexity of a string (like a DNA sequence) is the length of the shortest computer program that can produce that string. A random sequence has high complexity (no shorter description than itself), while a highly ordered sequence (like "AAAA...") has low complexity (can be described succinctly). Biological sequences and systems are neither random nor trivial; they have structure, patterns, redundancies, and modularity. AIT offers one way to try to quantify that structured complexity.

One challenge is that Kolmogorov complexity is uncomputable in general, and also context-dependent (it depends on the "reference universal Turing machine" chosen). However, researchers have been developing approximations and related measures that can be applied. For instance, the concept of **logical depth** (by Charles Bennett) distinguishes between randomness and organized complexity by considering how long a minimal program takes to run to produce the string. A random string has high complexity but shallow depth (no meaningful structure), whereas a highly organized string (like the digits of π or the genome of E. coli) might have high complexity and also high depth (it is compressible to some extent, but that compression is non-trivial – it encodes meaningful computations). In biology, one could argue that living systems are those with *both* high algorithmic complexity and high logical depth – they are not random, but they also cannot be described by a very short program (unlike, say, a perfect crystal which has low complexity).

Recent attempts to apply AIT include analyzing gene regulatory networks or protein interaction networks using algorithmic measures. As mentioned earlier, one study pointed out that "most information is in the connections among genes and not the genes themselves," suggesting that the network topology holds algorithmic complexity that a simple list of gene sequences misses ⁴¹ ⁴². To capture this, researchers like Hector Zenil have developed methods of approximating the algorithmic complexity of graphs (networks) rather than linear strings. They use techniques like lossless compression or algorithmic probability (estimating Kolmogorov complexity by searching for short programs that generate the graph). Such approaches have been used to compare the complexity of, say, the wiring of a neuronal network to a randomized control, or the structure of a metabolic network to alternative architectures ⁴¹. The results often confirm that biological networks are optimized in ways that reduce randomness (for robustness) yet are not overly regular (which would limit adaptability). In other words, life seems to sit in a sweet spot: not maximal entropy, not minimal entropy, but a poised state of **complex, context-dependent order**.

Another promising direction is using **AIT in origins-of-life** studies. Could we detect when a chemical system crosses from "random soup" to "algorithmically complex self-organization"? Sara Walker and colleagues have proposed measuring changes in the algorithmic compressibility of chemical reaction patterns or molecular distributions as a potential signal of life's onset. If a bunch of organic molecules can be significantly compressed (because they form repetitive patterns or functional motifs), that might indicate some generative process (like metabolism or templating) is producing non-random outputs. For instance, a test tube in which random polymers form will have a broad distribution of molecular weights and compositions (high entropy), whereas a test tube in which a self-replicating polymer has emerged will show

a skewed distribution (lots of copies of the replicator and related sequences – which is in a sense algorithmically simpler than a completely random set). By quantifying that difference, one might detect life's algorithmic footprint even without directly understanding the chemistry.

It's worth noting an intriguing link: some have suggested that **assembly theory and AIT are connected**. Indeed, a 2023 preprint formally demonstrated that under certain conditions, assembly index correlates with measures of compressibility (Shannon entropy of certain distributions, etc.) ⁴³ ⁴⁴. Intuitively, both are capturing the idea of *pattern richness*. Algorithmic complexity measures pattern in terms of description length; assembly theory measures it in terms of construction steps. Each unique pattern or subpattern that recurs in an object (like a repeating motif in a polymer) allows compression (in AIT) and also reduces the assembly index (because you can reuse that part rather than build from scratch). This convergence is encouraging: it means different approaches to quantifying life's complexity may be revealing the same underlying property.

Assembly Theory: Traces of Selection in Molecular Complexity

Assembly Theory (AT) is a fresh framework specifically developed to handle the question "what makes something likely the product of life, as opposed to chance?" Introduced by Lee Cronin, Sara Walker and collaborators, AT reframes objects (especially molecules) in terms of their *history*. Instead of treating a molecule as just a static structure with a certain entropy or complexity, AT asks: how many steps of combining building blocks are needed to make this molecule? This minimal number of steps is called the **assembly index (A)** ⁴⁵. A random process (like abiotic chemistry) might occasionally form a complex molecule, but the probability drops exponentially with the number of required steps. However, if selection is at play (as in life), it can produce highly complex objects in a reasonable time by iterative, biased processes. In other words, a molecule with a high assembly index is extremely unlikely to appear in abundance unless some evolving system (life) is systematically exploring molecular space and building it.

For example, consider two molecules: one is a simple organic acid, another is a complex lipid or alkaloid found in cells. The simple acid might be assembled in 2–3 reactions from basic precursors – low assembly index. The complex lipid might require 10+ specific steps – high assembly index. If you find a lot of the complex lipid in a sample, chances are it wasn't random chemistry; something (like a cell's metabolic network) was stringing together precursors in a guided way. Assembly theory thus gives an empirical handle: mass spectrometry can detect molecules and, through fragmentation patterns, infer their assembly index (roughly, how many sub-fragments it can be broken into). Remarkably, experiments have shown that samples from living matter (like bacteria extracts) contain molecules with assembly indices above a threshold rarely if ever seen in non-living samples ⁴⁶ ⁴⁷. This has been proposed as a biosignature test: send a mass spectrometer to Mars or Enceladus and look for molecules with A beyond, say, 15 – if you find them, life (or something like it) is the best explanation.

Beyond detection, assembly theory has deeper implications. It suggests a way to quantify **selection** itself. In Cronin and Walker's 2023 Nature paper, they argue that by considering objects in "assembly space," one can formally incorporate the causal history (selection processes) into fundamental physics ¹⁰ ³⁸. They frame it as not changing physics, but changing what we consider a fundamental object: not point particles, but these assembly patterns that carry a memory of the past. In assembly space, objects that were shaped by selection occupy a different region than those formed by chance. This offers a kind of bridge between biology and physics – making evolution "legible" to physics by encoding it in the complexity of objects.

Philosophically, AT posits that the universe allows *open-ended* creation of new complex objects (given the right processes like life), and it quantifies how much "causal effort" went into an object's existence 45 48.

There is active debate around assembly theory. Some critics argue that it's basically a rebranding of complexity measures that already exist, or that it conflates information with complexity in a way that could misidentify non-living phenomena as living (e.g., some crystals or human-made polymers might show high assembly index, though proponents would counter that context resolves that). Others welcome it as a much-needed practical tool for astrobiology and for quantifying when chemical systems become evolutionary. Regardless, it is a valuable attempt to **formalize a gradation between non-life and life**. We often assume life is a binary (either it is or isn't alive), but assembly theory suggests a spectrum: one could measure how far along something is in terms of assembly complexity. This might correspond to intuition (viruses, for instance, might sit in an intermediate range – products of life, but not alive themselves, they have high complexity but can't generate it autonomously).

In connecting AT to our earlier discussions, note how assembly index essentially captures the **imprint of an algorithmic process**. A random process doesn't create repeated building sequences (each outcome is like throwing dice anew), whereas a biological (algorithmic) process reuses sub-solutions and incorporates memory (each generation builds on the previous). The repeated substructures in molecules are like subroutines in code – they indicate a generative process with state. Thus, AT and AIT are aligned in spirit: life uses *algorithms* (in the broad sense) to generate complexity, and those algorithms leave statistical fingerprints.

Constructor Theory: Defining Life by Possible Tasks

Turning to a more theoretical angle, **Constructor Theory** (developed by David Deutsch and Chiara Marletto) is an emerging formulation of physics that shifts the focus from trajectories of particles to the *ability to perform tasks*. In constructor theory, the fundamental elements are "constructors" – entities that can cause transformations in other objects without being consumed (a catalyst is an example) – and the laws are expressed as which transformations are possible or impossible. The **Constructor Theory of Life** (Marletto 2015) attempts to capture what is special about life in these terms. Marletto concluded that for life (self-reproduction with open-ended evolution) to be possible under no-design laws (no teleology built into physics), the laws of physics must permit *information media* that can be reliably copied ⁴⁹ ¹¹. In essence, the only "non-trivial" requirement for life in constructor-theoretic terms is the existence of **digitally coded information that can act as a constructor** for itself when combined with some machinery (the "vehicle"). This is a rather elegant statement: it says that life will emerge in any universe where there is a medium (like DNA, or computer memory, or any stable set of states) that can act as a **recipe** and drive its own reconstruction through catalysts. The separation of replicator (information) and vehicle (the rest of the cell) echoes ideas from the 1970s (Dawkins' replicator-vehicle concept), but here it is derived as a requirement from fundamental principles.

What does this give us? It provides a potential **universal definition of life**: any system that contains a selfpropagating constructor that carries a program (information) for making itself is alive. This covers terrestrial life (DNA/RNA-based cells), and potentially could include alien life (maybe not nucleic acids, but some other digital polymer) or even artificial life (a self-reproducing robot with a blueprint). It excludes systems that don't have a mechanism for high-fidelity information copying. For instance, a simple autocatalytic chemical cycle that makes more of itself but without distinct information carriers might not qualify because it cannot undergo open-ended evolution – it will just keep making the same thing. Life, by this definition, requires a **heritable memory** (with variation) that leads to selection and evolution.

Constructor theory also addresses the long-standing question of **why life is not ruled out by physics**. Some early thinkers found life almost mysterious in a physical sense – how do highly ordered entities come about without violating thermodynamics? Constructor theory's answer: life is not miraculous; it's allowed because the laws of physics do permit certain complex tasks (like self-replication) to be performed, given the right catalysts. If the laws didn't allow stable information-bearing structures, life would indeed be impossible. But our universe has at least one such structure: sequences of nucleotides (and perhaps many more, since we discovered humans can engineer DNA-like polymers or other digital systems).

One might wonder how this abstract theory ties back to practical science. One payoff is conceptual clarity: it emphasizes that the core of life is *information that does something*. It's not just a pattern (a crystal is a pattern but doesn't do anything), and not just a process (fire is a self-sustaining chemical process but has no information memory). It's a synergy of information and process: a *program executed by a machine that produces that same program in new machines*. If we translate that to computation: life is like a program that prints its own code (with occasional mutations) and builds a new computer to run it. No ordinary software we write has that property (unless we intentionally make self-replicating code, like a virus). Life discovered that trick and, in doing so, unleashed an exponential growth and diversification we see in evolution.

The constructor perspective can also guide synthetic biology: it tells us that to create life from scratch, we must create a physical system where some configurations act as self-perpetuating constructors. This likely means getting a chemical system to the point of having a genetic subsystem (information) and a metabolic subsystem that together form a closed loop of construction. It's not enough to have a self-copying molecule; it must *construct* an entire self (which for a cell includes membranes, metabolism, etc.). Conversely, it implies that if we ever encounter something like a crystal that catalyzes its own growth, it wouldn't count unless it also encodes a blueprint for its own formation distinct from just being a template. Life's complexity in this view stems from the need to be a universal constructor (in von Neumann's sense) – containing a description and a means to copy that description and implement it.

In summary, these formal approaches (AIT, assembly theory, constructor theory) each reinforce the notion that life is characterized by **non-trivial information structure**. Life is not high entropy (random), but also not low entropy (crystalline order); it's *algorithmically structured*. Life produces objects of far greater complexity than unguided chemistry would ³⁸. Life persists through time by maintaining a chain of information that is copied and transformed but remains related (heredity). All these can be framed quantitatively and logically. We are beginning to see actual numbers and equations enter what was once purely philosophical territory. For example, assembly theory can assign a numerical assembly index to a molecule; AIT can estimate bits of algorithmic information in a genome; constructor theory can state life's criteria in logical postulates. This formalization is a sign of the field maturing. In coming years, we might see a convergence where these frameworks inform each other – perhaps a unified theory that tells us, say, *which* assembly indices correspond to the emergence of open-ended evolvability, or how the algorithmic complexity of a protocell's dynamics correlates with its fitness.

Discussion: Known Unknowns and Outlook for a Theory of Life

We have traversed a wide landscape – from cells as self-maintaining information processors, through the major evolutionary innovations that reconfigured life's computational capacities, to theoretical attempts to

capture life's essence in equations and measures. Yet, despite this progress, much remains unresolved. In this section, we discuss some open questions and challenges for the view of life as a computational/ informational phenomenon. These include both **known unknowns** – specific things we know we don't yet understand – and **potential unknown unknowns**, deeper gaps that current paradigms might not even fully recognize.

1. The Origin of Life – the gap between chemistry and computation: While we can outline scenarios, we still lack a definitive answer to how inert chemistry on the early Earth bootstrapped into the first self-replicating, information-bearing system. Is there a smooth continuum from autocatalysis to genetic code, or was there a hard threshold (a "spark" where information suddenly took over)? Experiments have made RNA molecules that can replicate themselves with some accuracy, but not enough to undergo open-ended Darwinian evolution. Could there have been intermediate systems – e.g., networks of mutually catalytic polymers – that gradually acquired information-like properties? This remains a frontier. The computational perspective suggests looking for the first appearance of **program-like behavior** in chemical systems. One testable hypothesis is that before fully autonomous genomes, there might have been chemical ecosystems in which information was distributed – like compositional genomes, where the relative concentrations of molecules carry hereditary information. Detecting or recreating such intermediate states in the lab is an ongoing challenge.

2. The nature of biological information – semantic meaning vs. syntax: Information theory treats information as bits devoid of meaning. But genetic information obviously *means* something to the organism (it encodes functional proteins, etc.), and neural information has meaning (representations of stimuli). How can we bridge the gap between the semantic aspect of biological information and the formal, syntactic measures? Some argue that meaning in biology arises from *function*: a sequence has meaning if it contributes to survival. Can we formalize that? Assembly theory takes a step by linking complex structure to a history of selection (function left a trace). Others have tried to quantify *functional information* – roughly, how much of the information in a sequence is actually used for a defined function ⁵⁰. For example, only certain mutations in a gene affect its function; the rest might be silent or redundant. The functional information would count just the crucial bits. This area is still developing. A breakthrough here would be being able to look at a genome or an ecosystem's information and quantify not just its complexity, but how much of that complexity is *meaningful* (propagates causally into the future of that system). This has implications for understanding things like junk DNA (is it truly junk or does it carry some subtle regulatory meaning?) and for astrobiology (could we tell a random polymer apart from one that encodes something useful?).

3. Reconciling analog and digital information: Biological systems use a mix of digital information (genes, discrete signals like action potentials in neurons) and analog information (metabolite concentrations, gradients, continuously varying membrane potentials). How do these interplay in life's computations? Digital systems have advantages for long-term memory (robust copying of genes), whereas analog systems are great for quick, nuanced responses (homeostasis, dynamic physiological regulation). The integration of the two is seen for instance in gene regulatory networks, where continuous signals eventually flip a digital-like switch for developmental decisions. One open question is whether a purely analog life form is possible or whether digitization (as in a genetic code) is a requirement for indefinite complexity. Constructor theory argues digital information is necessary for accurate replication (11). Is that a fundamental truth, or just a truth for Earth-life? Perhaps any complex life must evolve a code – an alphabet of some kind – to break away from the fuzziness of analog inheritance. This could be tested indirectly by studying if there are any

organisms (maybe hypothetical or simpler ones) that replicate without a genome-like template and still manage to evolve complexity (so far none known).

4. The limits of biological complexity: As life as a whole is an open-ended computational process, one can ask: are there limits to the complexity it can achieve? Some theorists speculate about a possible upper bound - maybe life cannot get arbitrarily complex because of fundamental constraints (energy availability per bit of information, error rates, finite resources, etc.). Others see no clear ceiling; humans might just be one step and future life (biological or AI or hybrid) could reach far higher levels of complexity and integration. Assembly theory implies that as long as selection continues, assembly index can keep growing, meaning more complex structures can keep appearing ⁴⁸. However, practical limits might intervene (e.g., if complexity growth outpaces error correction, systems might become unstable - there could be an error catastrophe at higher levels of organization too, not just at the gene level). Additionally, the evolution of intelligence and technology has created a situation where life's complexity is now expanding outside of the biological medium (into culture and machines). How do we account for that in our definitions? If an AI we create becomes self-improving and self-replicating, is that just a continuation of life's computational process in a new substrate? Many would say yes. This leads to almost philosophical questions about the destiny of life in the universe: is it to spread and maximize information processing (as some interpretations of the Fermi paradox suggest - maybe advanced life becomes invisible because it turns inward to computation)? These remain speculative but stem naturally from the view of life as computation.

5. Unknown unknowns – new principles awaiting discovery: Finally, we should acknowledge that our current frameworks might be missing something fundamental. Biology has surprised us repeatedly. For example, epigenetics revealed that heritable information is not just DNA sequence; cells transmit chromatin states and small RNAs across generations sometimes. Quantum biology is a nascent field examining whether quantum coherence plays non-trivial roles in things like photosynthesis or navigation in birds. If quantum effects are harnessed by life (still debated), does that add a layer to life's information processing that we haven't accounted for in our classical models? Perhaps life is even more computationally sophisticated, edging into quantum computation or other realms. Another possible unknown is the role of emergence at large scales – for instance, Gaia hypothesis type ideas where life as a whole planet behaves like a single self-regulating system. Is there "computing" happening at the ecosystem or biosphere level that is more than the sum of individuals (e.g., regulating climate within habitable bounds)? Some models indicate the biosphere does have self-stabilizing properties, but quantifying that as computation is extremely challenging.

We might also be missing what we haven't yet observed – *alien life*. All our theories are based on one example (Earth). Alien life might show information processing in forms we haven't conceived: maybe distributed in planetary magnetic fields, or using exotic chemistry that doesn't have discrete genes but something analogous. Our definitions need to be broad enough to encompass that, yet specific enough to exclude non-living phenomena. It's a delicate balance, and encountering a second sample of life (even microbial) beyond Earth would hugely inform the correctness of our computational thesis.

Conclusion: Toward a Unified Theory of Life's Information

In this narrative review, we have woven together insights from biology, physics, and information theory to bolster the thesis that **life is, at its core, a computational system – an information processing engine harnessed to a thermodynamic drive**. We have seen that life's defining characteristics can be reframed in informational terms: - **Self-maintenance (Autopoiesis)** – Living systems are self-generating networks that

use information (e.g., genetic instructions, feedback signals) to sustain their organization ² 51. - **Replication and Heredity** – Life propagates by copying information (genomes) with high fidelity, employing error-correcting mechanisms to preserve that information across time ⁷ ³. - **Functional Complexity** – Organisms exhibit multilayered complexity (molecular, cellular, organismal, ecological) arising from information integration across scales ¹⁴. Each higher level encodes and constrains information from lower levels, producing emergent properties. - **Adaptive Responsiveness** – Through computational-like sensing and control circuits, life responds to environmental inputs in pursuit of stability or growth (from simple chemotaxis in bacteria to complex cognition in animals) ⁶. - **Open-Ended Evolution** – Life's information, with selection feeding back to accumulate useful algorithms (biochemical or behavioral) in the population over time. This evolutionary process itself can be seen as a computation (a search algorithm in the space of possible organisms).

By grounding these concepts in literature and recent findings, we have strengthened the argument against the notion that "life as computation" is a mere metaphor. It is, in fact, *mechanistic*: life literally encodes information and performs computations (in the broad sense of transforming information to achieve function) as an inherent part of its operation. And it is *quantitative*: with tools like assembly theory and algorithmic complexity, we can start to measure just how much information processing is going on, and how life's complexity compares to other physical processes.

One of the salient points emerging from this review is the importance of **distinguishing different senses of computation** in interdisciplinary dialogues. When a biologist hears "the cell is a computer," they may recoil, imagining a reduction of life to a simplistic circuit diagram. But as we have clarified, what is meant is much richer: the cell is a *biochemical computer with self-fabrication capabilities*, nothing like your desktop PC yet performing analogous logical and control functions to achieve its self-perpetuation. As science progresses, these analogies are turning into detailed mappings. For example, the circuitry of the cell cycle (which controls cell division) has been mapped out in detail and shows switches, timers, and feedback loops that can be described with the mathematics of control theory and computation. In synthetic biology, researchers now design actual logic gates and oscillators using genetic parts, effectively programming cells to perform new computations (like detecting a cancerous state and releasing a drug). This is a profound confirmation that the computational view has teeth: if we can program life, it means we have understood some of its programming logic.

Another key takeaway is the unification of perspectives: **energy**, **information**, **chemistry**, **and evolution are all facets of one integrated narrative**. The origin of life required not just chemistry in the right setting, but also the emergence of information-bearing structures and a thermodynamic engine to run the first "programs" of life. Major evolutionary transitions, like the advent of eukaryotes or multicellular life, can be seen as upgrades to life's information processing capacity, often enabled by new energy regimes or new ways to manage entropy. Today, as we explore life-like behavior in other systems (e.g., computer viruses, AI, or complex automata), we continually use biological metaphors (we speak of computer "immune systems" or "evolutionary algorithms"). This is not coincidence – it reflects a deep parallel: *evolution is nature's algorithm for designing systems, and conversely algorithms we design can mirror evolutionary strategies*. We stand at a point where biology and computer science and physics are converging on common ideas of information and complexity.

What lies ahead? In practical terms, the computational perspective on life suggests several **future directions**: - In **origin-of-life research**, design experiments that track information-theoretic metrics (like

Shannon entropy, algorithmic complexity, assembly index) in replicating chemical systems. Success would be indicated by an inflection in these metrics, signifying a transition to information-driven growth ⁵² . If we can achieve that in the lab, it's akin to "seeing life's software boot up." - In astrobiology, refine informational biosignatures. As mentioned, assembly index is one; others might include detecting non-random patterns in atmospheric gases (information imprinted in planetary scale), or listening for structured signals. The computational theory of life suggests life anywhere will need some form of digital information system, so looking for evidence of such structures (e.g., polymers with repetitive motifs) is a promising path ¹⁰. - In biophysics, push stochastic thermodynamics further to identify fundamental limits: what is the minimum energy per bit for a cell to maintain its state? How close do cells get to Landauer's bound when replicating their genome or when firing neurons? Understanding these could lead to insights into why certain biological designs prevail (maybe cells evolved to be energy-information optimal). - In evolutionary theory, incorporate algorithmic models. Evolutionary biology has long used concepts like fitness landscapes; adding an information dimension (like considering the entropy of those landscapes or the computational complexity of reaching certain adaptations) could help explain phenomena like evolvability or stasis. The idea of "algorithmic phase transitions" might find other applications (are there points in evolution of complexity where progress stalls until a new representation or mechanism appears? Perhaps the origin of learning and brains was another such threshold). - In philosophy of biology, revisit definitions of life. Our review supports a definition along the lines of: Life is a self-sustaining chemical system capable of processing information to self-replicate and evolve 11. We might refine that to include the notion of hierarchical information (since even a virus processes information in some sense, but it is not self-sustaining without a host). The challenge is to be inclusive of weird life forms yet exclusive of non-life. The computational view helps by focusing on *what life does*, not just its components: life continuously performs computations that maintain its complexity. So perhaps any system that can compute its continued existence (predict and act to prevent its dissolution, and make offspring that do the same) should be considered alive.

In closing, the interdisciplinary synthesis presented here – spanning autopoiesis to assembly theory – is part of a broader trend: biology is increasingly a science of *information*. Just as 20th-century biology was revolutionized by discovering the molecular basis of genes (the structure of DNA, the genetic code), the 21st century is illuminating the informational basis of whole organisms and ecosystems. We are learning to read life's algorithms, edit them (with CRISPR and synthetic biology), and even design new ones. This review has attempted to narrate how our understanding of life's fundamental mystery – "What is life?" – is being reshaped by viewing living beings as computational systems of remarkable sophistication. In doing so, we directly addressed past criticisms by clearly distinguishing metaphor from mechanism and by substantiating claims with quantitative theory and empirical data. Far from being an oversimplification, the computational perspective is proving to be a unifying framework that connects disparate facets of life into a coherent picture ⁵³ ⁵⁴.

Life's story, then, can be seen as the story of information in the universe: how it emerges, self-organizes, perpetuates, and complexifies. It is a story that is still unfolding – both on Earth and, potentially, elsewhere. By continuing to explore life as a computational system, we not only inch closer to answering one of humanity's oldest questions, but we also stand to gain practical mastery over the processes that constitute living, potentially guiding the future evolution of life (biological and artificial) in a direction of our choosing. The ultimate test of this paradigm will be its ability to predict and perhaps *create* new life. If life is truly an algorithm, then in principle, once we understand it, we should be able to run it on purpose. Until then, life remains the most sophisticated "program" we know – one we are slowly learning to decode and, as we have reviewed, increasingly recognize as the computation that it is.