Journal of Theoretical Biology 537 (2022) 110984

Contents lists available at ScienceDirect

Journal of Theoretical Biology

journal homepage: www.elsevier.com/locate/yjtbi

An RNA-based theory of natural universal computation

Hessameddin Akhlaghpour

Laboratory of Integrative Brain Function, The Rockefeller University, New York, NY 10065, USA

ARTICLE INFO

Article history:
Received 5 November 2020
Revised 30 September 2021
Accepted 7 December 2021
Available online 31 December 2021

Keywords:
Turing-equivalence
Turing-completeness
Church-Turing thesis
Universal computation
Lambda calculus
Combinatory logic
Molecular engram
RNA
Junk DNA
Secondary structure

ABSTRACT

Life is confronted with computation problems in a variety of domains including animal behavior, single-cell behavior, and embryonic development. Yet we currently do not know of a naturally existing biological system that is capable of universal computation, i.e., Turing-equivalent in scope. Generic finite-dimensional dynamical systems (which encompass most models of neural networks, intracellular signaling cascades, and gene regulatory networks) fall short of universal computation, but are assumed to be capable of explaining cognition and development. I present a class of models that bridge two concepts from distant fields: combinatory logic (or, equivalently, lambda calculus) and RNA molecular biology. A set of basic RNA editing rules can make it possible to compute any computable function with identical algorithmic complexity to that of Turing machines. The models do not assume extraordinarily complex molecular machinery or any processes that radically differ from what we already know to occur in cells. Distinct independent enzymes can mediate each of the rules and RNA molecules solve the problem of parenthesis matching through their secondary structure. In the most plausible of these models all of the editing rules can be implemented with merely cleavage and ligation operations at fixed positions relative to predefined motifs. This demonstrates that universal computation is well within the reach of molecular biology. It is therefore reasonable to assume that life has evolved – or possibly began with – a universal computer that yet remains to be discovered. The variety of seemingly unrelated computational problems across many scales can potentially be solved using the same RNA-based computation system. Experimental validation of this theory may immensely impact our understanding of memory, cognition, development, disease, evolution, and the early stages of life.

1. Introduction

It may be argued that computation is the most fundamental aspect of life. Any problem that involves converting inputs to outputs where the informational content – rather than the material content – defines the problem is a problem of computation. Some examples of computation in biology include: using vision to guide wing movement in insect flight, language acquisition in humans, decision-making in single-celled ciliates (Dexter et al., 2019; Gershman et al., 2021), and embryonic development, the decisional process of beginning with a single cell and coordinating across daughter cells to produce a complex finely-detailed three-dimensional structure. Even though the computations that occur in these settings are poorly understood, it is generally assumed that the mechanistic building blocks that carry out computation in biology have already been identified. In the domain of animal behavior, these building blocks are believed to be neurons and neural networks. And in the domains of cell behavior and embryonic development, they are thought to be chemical cascades, gene regulatory networks, non-neuronal electric signaling (Levin, 2014; McLaughlin and Levin, 2018), and signal transduction pathways. But the adequacy of these building blocks is disputable and is not rooted in the theory of computation. In fact, if we are to take the theory of computation seriously, it is reasonable to consider that there may exist a computation system that remains undiscovered in biology.

For every computation system, there are problems that it can solve and problems that it cannot. The set of problems that a system can compute is its *scope*. Measurement of scope is agnostic to how the system works; the components of the system can be analog or digital, discrete or continuous, stochastic or deterministic. Biological computation systems are no exception. They can be analyzed in the framework of the theory of computation (see Fig. 1). A system is said to be at least as powerful as another if the former can simulate the latter (i.e., solve all of the problems in the other's scope). For example, combinatorial logic circuits (i.e., circuits consisting of boolean gates) are equivalent to look-up tables; both systems are capable of solving any problem defined over finite input/

E-mail address: hessam@rockefeller.edu

https://doi.org/10.1016/j.jtbi.2021.110984
0022-5193/

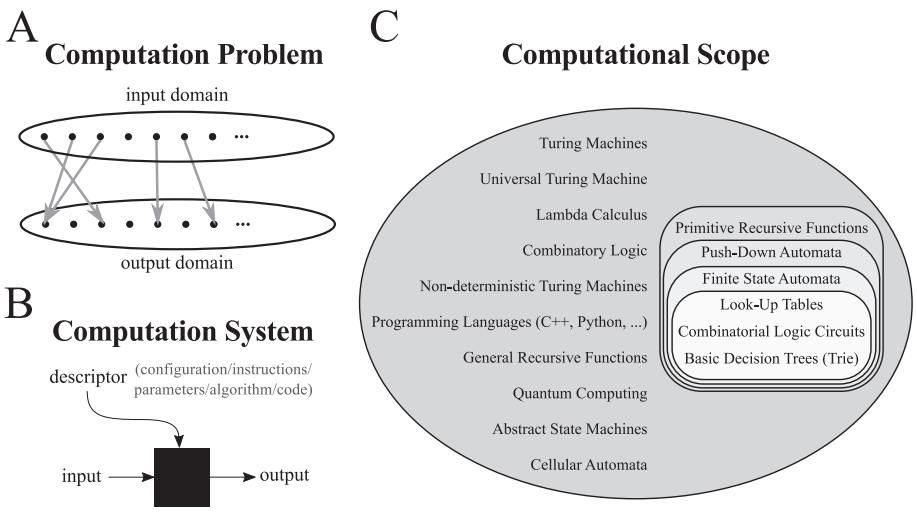

Fig. 1. *Basic Concepts from the Theory of Computation* **(A)** A computation problem is a partial mapping between an input domain and an output domain. For instance the problem of squaring a number is a mapping from numbers to their squared values. **(B)** A general-purpose computation system must be able to solve – not just one, but – many problems. Which problem it is to solve is determined by its *descriptor*. The *descriptor* may specify program instructions or the physical configuration that determines how the system works. For example, in a neural network computation system the description is the set of nodes, connections, weights, activation functions, etc. **(C)** A diagram depicting the computational scope of various systems. Computation systems very often emulate one another. The inner-most scope is the weakest form of computation and the outer scope includes all universal (i.e., Turing-equivalent) computers. There are many other scopes not shown in the diagram. These computation systems are all "transducers" rather than "recognizers"; their output domains are strings or natural numbers rather than binary values (i.e., accept/reject).

output domains. Finite state automata are strictly more powerful than either of those systems and can solve problems that are defined over infinitely large input/output domains (e.g., the problem defined by "given an arbitrarily large number as a string of digits, return the remainder of that number when divided by 7"). But there are some problems that finite state automata cannot solve (e.g., the problem defined by "given a string of open/close parentheses, determine whether it is balanced"). (See Boker and Dershowitz, 2005; Boker and Dershowitz, 2006; Boker et al., 2008 for a framework for comparing systems that operate on different domains).

One of the most profound mathematical discoveries of the 20th century was that there is a fundamental limit to computational scope (Church, 1936; Dershowitz and Gurevich, 2008; Kleene, 1936; Turing, 1936; Turing, 1937; Turing, 1937), meaning that there are some definable problems that are uncomputable; no effective computation system with finite means can solve them (see Appendix A for a discussion on what is meant by "effective computation" or "finite means" and why this is relevant to biology). Many abstract systems achieve that scope of computation. Such systems are said to have *universal* scope and are able to compute any computable function. Turing machines are one such system. There are many other universal computation systems, some of which were developed independently and do not resemble Turing machines in any obvious way. All of the universal computation systems listed in Fig. 1C are capable of simulating one another, but it is common to define universal computation by referring to Turing machines. A computation system is universal, or "Turing-equivalent", if and only if it can simulate any Turing machine.

1.1. The universal computer is missing in biology

Finite-dimensional dynamical systems are the dominant computational paradigm in biology, encompassing neural networks, biochemical signaling cascades, and gene regulatory networks. In these models the state of the system is described in terms of a number of physical quantities (such as membrane potentials of different cells or phosphorylation rates of certain proteins, or binding occupancy at various DNA sites). These quantities can positively or negatively influence one another. Network models are special cases of dynamical systems where the variables are represented by nodes and the interdependencies are sparse and can be drawn as a network. But physically relevant finite-dimensional dynamical systems are not known to be capable of universal computation. This claim may appear to contradict common wisdom since it was shown in the early 1990s that dynamical systems can simulate Turing machines (Moore, 1990; Moore, 1991; Siegelmann and Sontag, 1991). Based on those results it has been incorrectly asserted that chemical and neural networks are Turing-equivalent (Hjelmfelt et al., 1991; Sterling and Laughlin, 2015; Cabessa and Song, 2019). But in every instance where a finite-dimensional dynamical system has been shown to be capable of simulating Turing machines, that system lacked *structural stability* (Asarin et al., 1995; Branicky, 1995; Cabessa and Song, 2019; Fages et al., 2017; Koiran et al., 1994; Koiran and Moore, 1999; Moore, 1990, 1991; Siegelmann and Sontag, 1991; Šíma and Orponen, 2003; Graça et al., 2005; Pérez et al., 2019; Reif et al., 1990). Structural instability renders a system physically unrealizable. It means that even if the system is noiseless and the internal state variables have infinite precision, any arbitrarily small amount of error in the differential equations will lead to a radically different system that does not resemble the intended dynamics. (Structural stability is a distinct concept from chaos; chaotic systems are realizable and physically relevant).

Moore, who was the first to show that finite-dimensional dynamical systems are capable of simulating Turing machines (Moore, 1990; Moore, 1991), argued that *structural stability* is a reasonable criterion for determining whether a dynamical system may be possible to build or found to occur in nature and he conjectured that "finite-dimensional dynamical systems that are structurally stable (or generic by any other reasonable definition) are incapable of universal computation" (Moore, 1998). Moore's conjecture still stands today and we do not know of any biologically plausible network model capable of universal computation. (See Appendix B for a discussion on computational power of physically relevant dynamical systems).

Another common misconception is that the implementation of a universal logic gate (e.g. NAND or NOR) is sufficient for universal

computation (i.e. Turing-equivalence) (Magnasco, 1997; Scarle, 2009). It is quite easy to construct logic circuits with already identified bio-molecular building blocks (Benenson, 2012). But combinatory logic (not to be confused with "combinatory logic") is the weakest of the computation systems in Fig. 1C.

But why would life need a universal computer? Or what advantages might it have to be favored by natural selection? Perhaps living organisms get by without universal computation and the currently conceived network models are sufficiently powerful to address the problems and opportunities they face. This proposition is difficult to accept given the richness and complexity of life, particularly given how simple it is to achieve universal computation. It is not uncommon to accidentally stumble upon universality in systems where memory usage can expand. Notable examples are Conway's game of life (Rendell and Adamatzky, 2002), Wolfram's Rule 110 (Cook, 2004), Wang Tiles (Lafitte et al., 2008), and Schönfinkel's/Curry's combinatory logic (Cardone and Hindley, 2006; Schönfinkel, 1924). A powerful computing device would immensely benefit organisms that struggle to survive and reproduce. We know evolution is capable of designing remarkably sophisticated systems according to principles of optics, mechanics, chemistry, and thermodynamics. So why not principles of computation? There is nothing about universal computers that make them generally more costly, less efficient, or harder to maintain in comparison to weaker computation systems. In fact, the history of human technology suggests quite the opposite; analog non-universal systems are increasingly being replaced by digital microprocessors in devices and machines even though they do not strictly need universal computation for their purposes. (Microprocessors implement the von Neumann architecture which is a universal computation system capable of simulating any Turing machine).

Let us entertain the possibility that a universal computer exists in biology but has not yet been found. The most obvious principle that may guide us towards finding such a system is *memory expansion*. A necessary [but not sufficient] condition for universality is that memory usage (in systems where "memory usage" can be defined) not be bounded by the system's descriptor (see Fig. 1B). In other words, the system should be able to recruit more memory space when needed *during* the process of computation or *after* it is given an input. (Memory usage in finite state automata, a weaker non-universal computation system, is bounded by the system's descriptor irrespective of the input). In the context of neuroscience and cognition, the importance of separating memory from computation and the need for a read-write mechanism has been explicitly raised by Gallistel & King as a critique of the network paradigm (Gallistel and King, 2009).

Network models cannot easily be reconciled with the memory expansion condition. The solution adopted in the Turing-equivalent dynamical systems discussed above is to expand memory by using the numerical digits of a variable as a string of symbols (Moore, 1990; Moore, 1991; Siegelmann and Sontag, 1991). This method has a severe practical limitation; realistically, less than a few dozen bits of memory can be recruited and, more importantly, it leads to structural instability which renders it impossible to build or find in nature. Another solution is to assume that the network can grow in its number of variables. This can be achieved by either having the system physically grow through the construction of new components during computation (e.g., by the generation of new neurons or creation of entirely new molecules/genes according to specific rules) or by assuming there is an arbitrarily large reservoir of *silent* or *dormant* dimensions (e.g., implemented by repetitive network architectures) that serve as general purpose units of memory and can be accessed for storage and retrieval. Crucially, the number of memory units should not be part of the system's *descriptor* that determines the problem that is supposed to be solved.

There has been notable progress in recent years toward building network models that use architectural motifs (instead of numerical digits of a physical quantity) as a memory tape. Papadimitriou et al. implemented the memory tape component of Turing machines using neuronal assemblies, but an exogenous agent was needed to carry out the logic of the Turing machine's tape head (Papadimitriou et al., 2020). Graves et al. implemented the Turing machine's tape head as a neural network but their memory tape was a not implemented with a neural network (Graves et al., 2014). It remains to be shown whether a fully neuronal model can combine all the components of a Turing machine together in a biologically plausible manner.

Another potentially promising direction is to search for implementations of cellular automata in biology (Ermentrout and Edelstein-Keshet, 1993). Cellular automata are potentially universal in scope; depending on the transition rules, it may be possible to use them to compute any computable function (although they may incur non-linear slow-downs relative to Turing machines). For instance, Oku & Aihara modeled a neural network implementation of Rule 110, a Turing-equivalent linear cellular automata (Oku and Aihara, 2010). (What makes this model different from standard neural network models is that the size of the network is not bounded by the descriptor; the descriptor can be defined to only specify the initial activity pattern without specifying the full network that that pattern is embedded in). Alternatively, reaction–diffusion systems can be used to implement cellular automata in a non-neural system (Baluška and Levin, 2016). A natural implementation of cellular automata presupposes repeating motifs (e.g. biological cells, or repeating network architectures) which contain activity patterns that are invariant to spatial shifts; the evolution of an activity pattern in time should not depend on where that pattern is located in the substrate. This is a key property of cellular automata that allows it to fulfill the memory expansion condition. A pattern can recruit more memory units by expanding in the surrounding space and every computational program can be described independent of the physically available memory capacity. It is yet to be determined whether cellular automata bear any utility in understanding how biology might achieve universal computation.

Perhaps the most promising place to search for a universal computer is in the molecular biology of polynucleotides. Memory expansion is trivial in a system that uses the precise sequence composition of polynucleotides as memory. It can be accomplished through the addition of nucleotides, either by insertion or tail extension. The resemblance of polynucleotides to strings of computation theory is hard to ignore. In Turing's attempt to formalize the notion of computation he wrote: "*Computation is normally done by writing certain symbols on papers... I think that it will be agreed that the two dimensional character of paper is no essential of computation. I assume then that the computation is carried out on a one-dimensional paper,* i.e., *on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite... The effect of this restriction on the number of symbols is not very serious. It is always possible to use sequences of symbols in the place of a single symbol*" (Turing, 1936).

Polymer sequences consisting of an alphabet of only four nucleotide symbols A, C, T (or U), and G elegantly fit this description. This striking resemblance was perhaps first noticed in the 1970s by Bennett who later proposed a blueprint for a RNA/DNA based Turing machine (Bennett, 1982; Bennett, 1973). Apart from their string-like structure, polynucleotides possess many properties that make them ideal vehicles for biological computation, e.g., thermodynamic stability, spatial compactness, and their capacity to be modified with low energy cost (Gallistel, 2017;

Langille and Gallistel, 2020). It is not surprising that these molecules are considered to be potentially useful in data storage technology (Ceze et al., 2019; Grass et al., 2015) and biomolecular computing (Benenson, 2012; Ruben and Landweber, 2000; Chen and Wood, 2000). DNA/RNA computation has been used in laboratory settings to solve a number of challenging tasks including the Hamiltonian path problem (Adleman, 1994), the travel salesman problem (Lee et al., 2004), the boolean satisfiability problem (Lipton, 1995; Liu et al., 2000; Sakamoto et al., 2000), and the knight placement problem (Faulhammer et al., 2000).

Already, a number of DNA/RNA-based computation models have been proposed that are universal in their computational scope. One approach is to have DNA strands self-assemble into two-dimensional structures that obey the laws of Wang tiles or linear cellular automata (Mao et al., 2000; Rothemund et al., 2004; Winfree et al., 1998; Woods et al., 2019). In this approach, memory expansion is achieved through the growth of an aperiodic crystal. Another general approach is to use DNA strand displacement to build a chemical reaction network (Cardelli, 2011; Soloveichik et al., 2010; Soloveichik et al., 2008; Qian et al., 2011). It has been suggested that chemical reaction networks can be used to implement counter machines (which are Turing-equivalent) by relying on the exact number of molecules that are present (Soloveichik et al., 2008). A third theoretical approach is to use DNA strand displacement to implement stack machines or Turing machines (Qian et al., 2011; Yahiro et al., 2016; Lakin et al., 2011), where memory expansion is achieved by the lengthening of the DNA strand at its ends. A fourth approach is to enzymatically modify the content of a DNA strand that serves as a Turing machine memory tape (Bennett, 1982; Shapiro, 2012; Shapiro and Karunaratne, 2001; Varghese et al., 2015). Applying this same approach to chromatin can yield a chromatin computer where DNA and histone modification rules can implement a Turing-machine (Bryant, 2012). A more recent approach is to exploit the complexity of folding pattens of RNA molecules during transcription (Geary et al., 2019; Geary et al., 2017). Rothemund's approach, which may be the earliest detailed Turing-equivalent model, was based on restriction enzymes acting on a circular double stranded DNA, where complementary DNA overhangs are ligated (Rothemund et al., 1996).

Although one of the goals of DNA/RNA based computation has been to implement it in living cells (Shapiro and Gil, 2008; Win and Smolke, 2008; Shapiro and Benenson, 2006; Siuti et al., 2013; Benenson, 2009; Benenson, 2009), it would be hard to make the case that any of these Turing-equivalent models already exist in nature as a general purpose computation system. The above body of work has been design-oriented, not discovery-oriented (perhaps with the exception of Bryant, 2012). The question I would like to raise here is not whether it is feasible to artificially implement a universal computation system using polynucleotides, but whether it is reasonable to consider that nature may have already done so. This question will lead us to an RNA-based model that looks very different from the ones that have been proposed to date.

Recent developments in molecular biology suggest the possibility that non-protein-coding RNA have a yet undiscovered critical role. Approximately 1.74% of the human genome ends up in mature mRNA and more than half of that consists of untranslated regions that do not encode proteins (Piovesan et al., 2016). This is while it is has been found that the vast majority of the human genome is actively transcribed (ENCODE Project Consortium et al., 2007; Kapranov and St. Laurent, 2012; Clark et al., 2011). The discovery of pervasive transcription was met with controversy (Kapranov and St. Laurent, 2012; Clark et al., 2011; van Bakel et al., 2011; Cheng et al., 2005; Dinger et al., 2009; Kapranov et al., 2010; Palazzo et al., 2014) and the functional significance of the non-coding portion of the transcriptome is being intensely debated (Palazzo et al., 2014; Doolittle, 2013; ENCODE Project Consortium, 2012; Freedman et al., 2011; Graur et al., 2013; Lee et al., 2019; Linquist et al., 2020; Mattick and Dinger, 2013; Niu and Jiang, 2013; Palazzo and Lee, 2015; Pheasant and Mattick, 2007). Proponents of the "junk DNA" hypothesis estimate that no more than 15% of the human genome can have functional significance, and the rest leads to "transcriptional noise" when transcribed (Graur, 2017; Ponting and Hardison, 2011; Rands et al., 2014). The opposing viewpoint argues that most of the human genome may be functional and that sequence conservation is not a necessary condition for functional relevance (Lee et al., 2019; Mattick and Dinger, 2013; Pheasant and Mattick, 2007; Aprea and Calegari, 2015) and that there are many other indicators of function such as conservation of secondary structure (Smith et al., 2013; Washietl et al., 2005), conservation of promoters sequences (Derrien et al., 2012; Guttman et al., 2009; Kutter et al., 2012; Ponjavic et al., 2007; Stephen et al., 2008), cell-specificity in expression levels (Derrien et al., 2012; Cabili et al., 2011; Gloss and Dinger, 2016; Ravasi et al., 2006), subcellular organization (Mercer et al., 2008; Sone et al., 2007), and temporal regulation during embryonic development (Li et al., 2020; Pauli et al., 2012). Genome-wide association studies show that more than 70% of the genetic loci associated with traits and diseases fall in intergenic or intronic regions (Freedman et al., 2011). These regions have been found to be abundantly transcribed (Bartonicek et al., 2017; St. Laurent et al., 2014) in a highly cell-type specific manner consistent with their associated traits (Hon et al., 2017). Across organisms the non-protein-coding to protein-coding ratio of the genome scales with organism complexity, while the number of protein-coding genes as well as the total length of protein-coding sequences plateaus (Liu et al., 2013; Mattick, 2004; Taft et al., 2007). While there are many indicators suggestive of function, the mechanistic roles of non-coding RNA remain to be discovered. The question of non-coding RNA function has even been described as "*the most important issue in genetics*" (Kapranov and St. Laurent, 2012). I propose the theory that the non-protein-coding portion of genome and transcriptome contains the data and programming material of an undiscovered universal computation system in biology.

2. Natural RNA-based universal computation is plausible

In support of the theory of natural universal computation through polynucleotides I demonstrate that universal computation through RNA is in principle attainable without assuming extraordinarily complex molecular machinery. A molecular machine that implements a universal Turing machine would be extraordinary and implausible, as it would require large enzymes operating in a far more elaborate manner than the ribosome. Even if such a system were molecularly feasible, it is hard to imagine how it could have gone undetected. I present an alternative class of models based on λ-calculus and combinatory logic. Computation in these models is a decentralized process where distinct enzymes make local modifications to RNA molecules according to just a handful of editing rules. The specific details of the models are somewhat arbitrary and only meant to be used as a proof a principle, demonstrating that it is possible to implement a universal computation system through basic molecular operations on RNA. I argue that the models are plausible and that it is conceivable such a system may have evaded detection throughout the many decades of research in molecular biology.

2.1. Combinatory logic and λ-calculus as computation systems

λ-Calculus and its predecessor, combinatory logic (CL), are two nearly identical universal computation systems. The entities defined in these systems are functions that take functions and return functions. No distinction is made between programs and data and everything is constructed as a function. There is a one-to-one equivalence between lambda functions (called "λ-terms") and combinatory functions (called "combinators"). The difference lies in the elementary operations that are used to compute things. λ-calculus computes using variable substitution and variable renaming. CL uses applications of primitive combinators. (I will only briefly introduce CL here. For a more complete introduction to both systems see Appendix C).

The identity combinator **I** is defined as **I***x* = *x*. It returns whatever it is given. The combinator **K** is defined as **K***xy* = *x*. It takes two arguments and returns the first. **C** takes three inputs and swaps the second and third, **C***xyz* = *xzy*. **B** is defined as **B***xyz* = *x*(*yz*), **W** is defined as **W***xy* = *xyy* (see Fig. 2A for more primitive combinators). (Combinators are capitalized and variables are in lowercase). Let us evaluate the term **CIBW**. At every step we apply the left-most combinator. Applying **C** results in **IWB**. Next we apply **I** to get **WB**, which cannot be evaluated further because application of **W** requires two inputs. So the final result is **WB**. Let us try another example, this time with a variable: **CKC***x* = **K***x***C** = *x*. Like the **I** combinator, **CKC** returns *x* given any *x*. This shows that **I** can be constructed using **C** and **K**. Finally, let us evaluate an example with parentheses: **BCKIW(KK)KC**. First we apply **B** to get **C(KI)W(KK)KC**. To apply **C** we swap the second and third arguments, **W** and **(KK)**, to get **(KI)(KK)WKC**. We can always remove the parentheses around the left-most term because CL is by convention left-associative. Doing so, we get **KI(KK)WKC**. Applying **K** we get **IWKC** = **WKC** = **KCC** = **C**. So **BCKIW(KK)KC** = **C**.

Remarkably, the set of primitive combinators **B**, **C**, **K**, and **W**, constitute a universal computation system. Using only these four combinators, it is possible to simulate Turing machines and compute any computable function. Not only is it possible to compute boolean logic circuits (see Fig. 2B), but CL can implement data structures and recursive algorithms (see Appendix C). For example **C**(**C**(**B**(**BK**))*z*)(**C**(**C**(**B**(**BK**))*y*)(**C**(**C**(**B**(**BK**))*x*)(**KI**))) can be interpreted as a stack containing three elements x, y, and z. And **B**(**WI**)(**BWB**)(**B**(**B**(**C**(**B**(**B**(**B**(**C**(**BC**)**I**)(**BC**(**CI**)))(**BC**)))(**BC**(**C**(**B**(**BK**)))))**B**)(**KI**) is a recursive program that takes a stack of any size and reverses the order of its elements. (To see how I constructed these terms see Appendix C). CL can implement numbers and arithmetics. The most popularized number systems are unary (e.g., Church numerals). This has given λ-calculus and CL a reputation for being slow. But it is not difficult to implement efficient arithmetic operations with binary numeral or in any base of choice (Mogensen et al., 2001). λ-calculus and CL are as powerful and expressive as any functional programming language, and can simulate Turing machines with linear slow-down (see Appendix C for proof).

2.2. Combinatory logic can be implemented through RNA editing rules

In CL, terms are usually represented as strings of characters. Nucleotide sequences can trivially be used to represent strings. There is already a precedent for this in protein coding sequences where each triplet represents an amino-acid. Similarly, each primitive combinator can be encoded using sequence motifs. (Open and close parentheses can also be represented by unique motifs, but this is not the method used in the models I present below.) Not all nucleotide sequences must represent a combinator (some can be neutral fillers) and combinator motifs need not be unique (there may be redundancy similar to amino-acid codons). A small enough primitive combinator set, like S and K, can make it possible to use just one nucleotide per combinator. Non-canonical bases and nucleotides modifications like methylation may also be involved in the representation scheme. I refrain from speculating over the motifs for the combinators, but I only remark that if the codes for the primitive combinators are all of the same length, it can make molecular implementation of the combinator rules simpler.

To evaluate a term, it is sufficient to recursively apply the left-most combinator, as illustrated in the examples above, until it is no longer possible to reduce the term. This can be accomplished by distinct and independent enzymes, each responsible for implementing one of the primitive combinators. (The enzymes need not be proteins; they can be other RNA strands or even self-cleaving/self-ligating RNA elements). For example, if there are four primitive combinators there can be four distinct and independent enzymes that each apply one of the combinators by first recognizing the left-most motif that encodes for it and then applying appropriate changes to the RNA strand. These applications can be carried

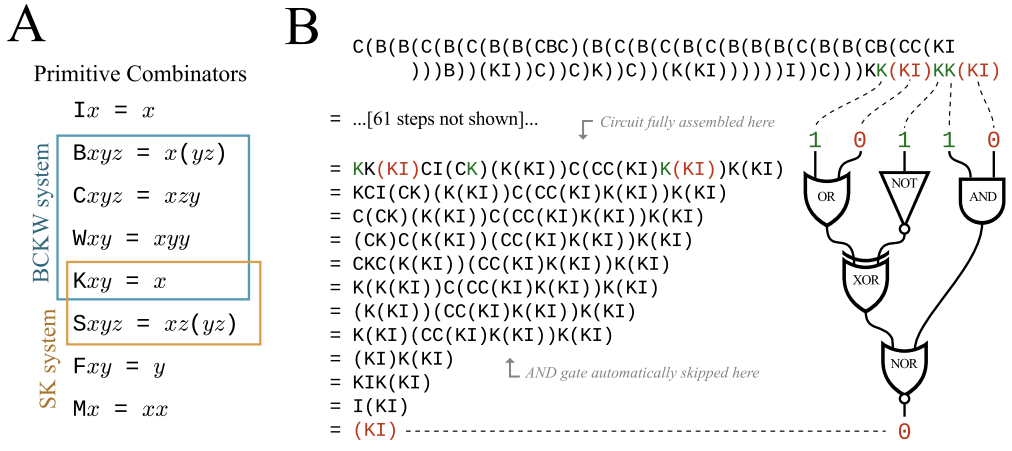

Fig. 2. *Implementation of a combinatorial logic circuit using combinatory logic* – **(A)** Definitions of some commonly used primitive combinators. Outlined are two common universal systems: SK and BCKW. **(B)** A combinatory term was constructed using combinators B, C, K, and I to compute a circuit with five boolean inputs. Here, Church's boolean encoding is used, where K represents *true* and KI represents *false*. At each step, either the parentheses enclosing the left-most term are deleted or the left-most combinator is applied. The 61 omitted steps only rearrange the five inputs and assemble the circuit. Evaluation of the AND gate was automatically skipped since the first input to the NOR gate was true. The final result is KI, equivalent to false. Combinatory logic must not be confused with combinatorial logic. The latter is the weakest of the systems in Fig. 1C (equivalent to look-up tables), whereas the former is a universal computer (equivalent to Turing machines).

out in an uncoordinated fashion and there is no need for a central entity to direct these operations.

If we take nucleotide sequences to represent CL terms, it seems more reasonable to assume that they are parsed and edited at the 3′-end. The 5′-end is typically capped and protected whereas the 3′-end is highly dynamic and amenable to additions or removals of nucleotides; RNA synthesis occurs by polymerization at the 3′-end and the poly(A) tail (a stretch of up to 200 adenine bases at the 3′-end) can lengthen or shorten even after an RNA is exported to the cytoplasm (McFleder et al., 2017; Udagawa et al., 2012). Additionally, RNAs exhibit high diversity in their 3′ splice site. The same RNA transcript can be cleaved and polyadenylated at different sites depending on the context (Elkon et al., 2013; Flavell et al., 2008). The variability of the 3′-end is especially pronounced in the brain where 3′UTRs of mRNA are lengthened beyond the annotated ends (Guvenek and Tian, 2018; Miura et al., 2013) and where miRNA show extensive sequence modifications at their 3′-end (Martí et al., 2010). For these reasons I will assume that RNA strands are parsed as combinatory terms in the 3′-to-5′ direction. Respecting the standard nomenclature, I refer to the direction of the 5′-end as "upstream" and the direction of the 3'-end as "downstream".

What sort of RNA modifications are required to implement combinator applications? The answer depends on the primitive combinator set. **B**, **C**, **K**, and **W** are sufficient for universality. So is the smaller set of only two combinators **S** and **K**. (All four combinators **B**, **C**, **K**, and **W**, can be constructed using only **S** and **K** – see Appendix C). There are infinitely many valid basis sets that would produce a universal computation system. Fortunately, we can evaluate the plausibility of the model without assuming the exact basis set. For any set of primitive combinators to achieve universal computation, it must have at least one combinator that deletes terms (**deletion**), at least one that reorders terms (**reordering**), at least one that duplicates terms (**duplication**), and at least one that adds parentheses (**nesting**). (This is easy to prove by showing that none of these operations can be mediated by the other three. Without deletion the number of combinators cannot shrink. Without duplication the number of combinators cannot grow. Without reordering the ordering of terms remains invariant. And without nesting the number of parentheses – in, say, left-associative representations – cannot grow). In the **BCKW** system, each of the four combinators fulfills exactly one of the four conditions. In the **SK** system, the **S** combinator fulfills the last three conditions and the **K** combinator fulfills the first. We can examine the plausibility of each of these four operations separately.

Deletion: The enzyme responsible for implementing the deletion operation needs to excise a segment of RNA. Splicing is one of the most common RNA modifications and many enzymes are known to mediate it. Splicing involves cleavage and ligation but even a single cleavage operation is sufficient to completely fulfill the deletion condition. For instance, the **F** combinator, equivalent to Church boolean *false*, is a combinator that takes two inputs and returns the second (**F***xy* = *y*). An enzyme responsible for implementing **F** only needs to cleave the RNA one term upstream of the **F** motif (corresponding to deletion of "Fx" in "Fxy"). (The **K** combinator, defined as **K***xy* = *x*, can be constructed as **K** = **CF** and no longer needs to be in the basis set). RNA strands that are excised through this method must be immediately discarded and not interpreted as representing combinatory terms. This can be trivially implemented since excised strands lack a 5'-cap and are susceptible to exonuclease degradation.

Reordering: The enzyme responsible for implementing the reordering operation only needs to conduct something as simple as swapping two elements corresponding to successive terms, e.g., as in **C***xyz* = *xzy*. Similar RNA modifications are already known to occur in cells. Post-transcriptional re-ordering of exons was first observed in the early 1990s (Cocquerelle et al., 1992; Nigro et al., 1991). At first, it was thought to be rare, expressed at low levels, and confined to circular RNAs. But several recent studies suggest that it may occur abundantly, occurring in polyadenlyated transcripts at expression levels comparable to that of their canonically spliced counterparts (Al-Balool et al., 2011; Dixon et al., 2005; Hamilton, 2012; Horiuchi and Aigaki, 2006; Kong, 2005; Shao et al., 2006). Transposable elements in DNA, RNA's sister molecule, frequently move around changing locations with the help of transposase enzymes that mediate their cleavage and ligation. Operationally it only requires three cleavages and three ligations to fulfill the reordering condition, both of which many native enzymes like the spliceosome are capable of (Al-Balool et al., 2011). It is therefore plausible to assume other enzymes may exist that are capable of reordering RNA elements.

Duplication: Duplication is not as trivial as deletion or reordering. Example combinators that require duplication are are **W***xy* = *xyy* and **M***x* = *xx*. The term that needs to be duplicated may either be a motif for a single combinator or a nested (i.e., parenthesized) sequence of arbitrary length. Duplication of arbitrarily long sequences requires an enzyme that can synthesize a copy of an RNA element. RNA dependent RNA polymerases (RdRp), the enzymes that can directly synthesize RNA from an RNA template, are common in viruses and have also been found in plants and nematodes. But many species including humans and fruit flies lack endogenous RdRp. Are there any known methods of RNA sequence duplication that may exist in all cells? One method is through reverse transcription (i.e., DNA synthesis from an RNA template) followed by transcription, (RNA synthesis from a DNA template). Reverse transcriptase and RNA polymerase, exist abundantly across the plant and animal kingdom, although, most bacteria species lack reverse transcriptase (Lim and Maas, 1989; Simon and Zimmerly, 2008) and reverse transcriptase is thought to be inactive in many – if not most - cells of multicellular eukaryotes.

But a more promising candidate enzyme for mediating sequence duplication in the model is RNA polymerase (RNAP). RNAP is the crucial transcription enzyme, abundantly present in all living organisms, that normally synthesizes RNA from DNA templates. In special cases, RNAP can polymerize RNA solely from RNA templates (Lehmann et al., 2007). RNA replication through RNAP is the method used by viroids that infect plants, the hepatitis delta virus (HDV) that infects humans, and several other related RNA based viruses that lack their own polymerase enzyme (Chang et al., 2008; Chang et al., 2019; Lai, 2005; Tseng and Lai, 2009). These viruses rely on native RNAP in host cells for RNA replication. In the case of viroids and HDV, the replicated RNA is circular and replication is understood to happen through a rolling circle mechanism (Tseng and Lai, 2009; Flores et al., 2011). RNA replication of non-circular RNA strands has been demonstrated in vitro (Biebricher and Luce, 1996; Biebricher and Orgel, 1973; Jain et al., 2020; Kakimoto et al., 2015; Konarska and Sharp, 1989; Konarska and Sharp, 1990; Wettich and Biebricher, 2001) Remarkably, this mode of replication is dependent on the existence of a 3′-GG… or 3′-CC… motif on the template, begins synthesis immediately after the motif, and can generate concatemers from linear (non-circular) templates (Jain et al., 2020; Konarska and Sharp, 1989). This is already quite close to what would be expected of an enzyme responsible for implementing the **W** or **M** combinators. (It must first recognize the motif that codes for **W** or **M**, begin synthesis immediately upstream of the motif, and produce a strand with two copies of the term that is intended for duplication). At any rate, the hypothetical duplication enzyme may still be unknown to us and there is considerable evidence that direct RNA duplication through unknown mechanisms occurs endogenously (Dixon et al., 2005; Dixon et al., 2007; Frantz et al., 1999; Kapranov et al., 2010; Rigatti et al., 2004). (In the next section

we will revisit the problem of RNA duplication and show that it can be outsourced to DNA-based RNA transcription).

Nesting: A nesting operation composes two terms as a single nested term. For example, **B** defined as, **B***xyz* = x(*yz*), nests its second and third arguments as a single term (*xyz* is interpreted as ((x*y*)*z*) in left-associative CL). If parentheses are encoded by sequence motifs, as suggested earlier, the enzyme responsible for implementing **B** needs to insert predefined open and close motifs one and three terms upstream of the motif for **B**. But this method brings out a complication that we have ignored until now: parenthesis matching.

Parenthesis matching is critical for the model because all of the enzymes that apply combinators need to recognize and count whole terms. For example, if the hypothetical deletion enzyme is confronted with an open parenthesis at the position of a term that is supposed to be deleted, it must delete the entire stretch of nucleotides between that open parenthesis and its matching close parenthesis (which can include other open/close parentheses). How can an enzyme when confronted with the start of a term find the correct ending nucleotide? This is not a simple task. A parsing algorithm that implements parenthesis matching needs to keep track of the parenthesis depth, incrementing with every "(", decrementing with every ")", and stopping whenever the calculated depth reaches zero. A molecular implementation of this algorithm does not appear plausible; I cannot conceive of an implementation of this algorithm without invoking extraordinarily complex hypothetical molecular machinery. Fortunately, there is an elegant solution to this problem using the base pairing properties of RNA.

Similar to DNA, RNA molecules can form double stranded helices with their complementary sequences. Base pairing can also occur within the same strand. When an RNA molecule contains two sequences that are inverse complements of one another, those sequences physically come together to form a stem-loop (Fig. 3A). Stem-loops can occur inside other stem loops and the entire base-pairing organization of an RNA molecule is referred to as the secondary structure. RNA strands typically have intricate secondary structures that involve many layers of nested stem loops (Fig. 3B-D).

If open and close parentheses are represented by reverse complementary sequences, RNA molecules naturally solve the problem of parenthesis matching by physically bringing matching parentheses together in space. It is then enough for the combinator enzymes to treat the base of a stem as a single term, just as they would for a primitive combinator motif. For example, a hypothetical enzyme that implements **F***xy* = *y* must cleave one term upstream of the **F** motif. If a stem loop appears in the place of *x*, it can delete the entire stem loop by cleaving at the base of the stem. This model suggests a very general role for RNA secondary structure that is more fundamental than anything presently conceived (Wan et al., 2011).

In light of this solution, let us evaluate the plausibility of a nesting operation. To implement nesting, there must be a method of adding new stem loops into RNA strands. This can be done through insertion of RNA duplexes. Double stranded RNA (dsRNA) is known to exist in cells and its over-expression or under-expression can be lethal (Liddicoat et al., 2015; White et al., 2014). The enzyme responsible for implementing nesting may recruit these dsRNAs, or possibly recycle duplexes that have been removed in previous CL operations, and insert them in place. (A basic operation that the model needs is parenthesis removal. If the left-most term is enclosed in parenthesis, the parentheses should be removed. Removed duplexes can be reused in the same strand for nesting operations). Insertion of RNA duplexes has not been documented in cells, but it would only involve simple cleavage and ligation operations.

Now that we take stems to represent parentheses, it is possible to provide a set of concrete RNA editing rules that can, in theory, implement combinatory logic. The five rules depicted in Fig. 4 implement left-associative CL based on four arbitrarily chosen combinators. In left-associative CL *abcde* is interpreted as (((*ab*)*c*)*d*)*e*. A variant of these rules can be constructed by implementing

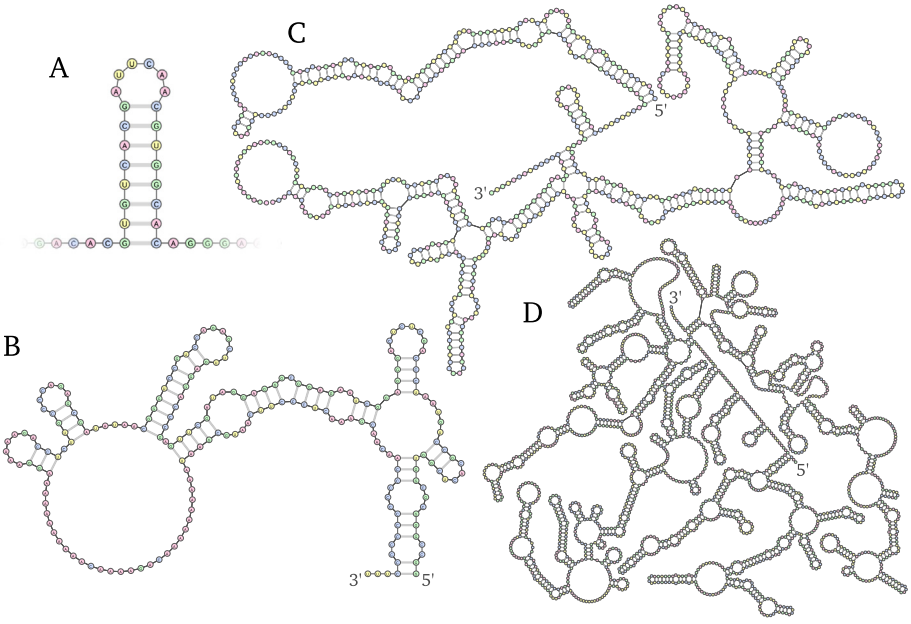

Fig. 3. *RNA Secondary Structure* – **(A)** RNA stem-loops form when the nucleotides of two segments of an RNA strand pair with one another. The region containing base-pairs is the stem and region in between the stem is the loop. Base pairing in RNA follows the canonical Watson-Crick rules where A pairs with U and G pairs with C but can also include less stable wobble pairs such as G-U. **(B-D)** Examples of secondary structure of some non-protein-coding RNAs exhibiting many levels of nested stem loops. **(B)** 200-nt BCYRN1 transcript, expressed in neuronal dendrites and implicated in memory loss and cancer (Samson et al., 2018). Secondary structure was obtained using RNAfold minimum free energy prediction (Mathews et al., 2004; Gruber et al., 2008; Lorenz et al., 2011). **(C)** 683-nt lncTCF7 transcript, implicated in cancer. Secondary structure was obtained from (Owens et al., 2019). **(D)** 2148-nt HOTAIR transcript, expressed in peripheral tissues and implicated in epidermal differentiation and development. Secondary structure was obtained from (Somarowthu et al., 2015). All figures were drawn using VARNA (Darty et al., 2009).

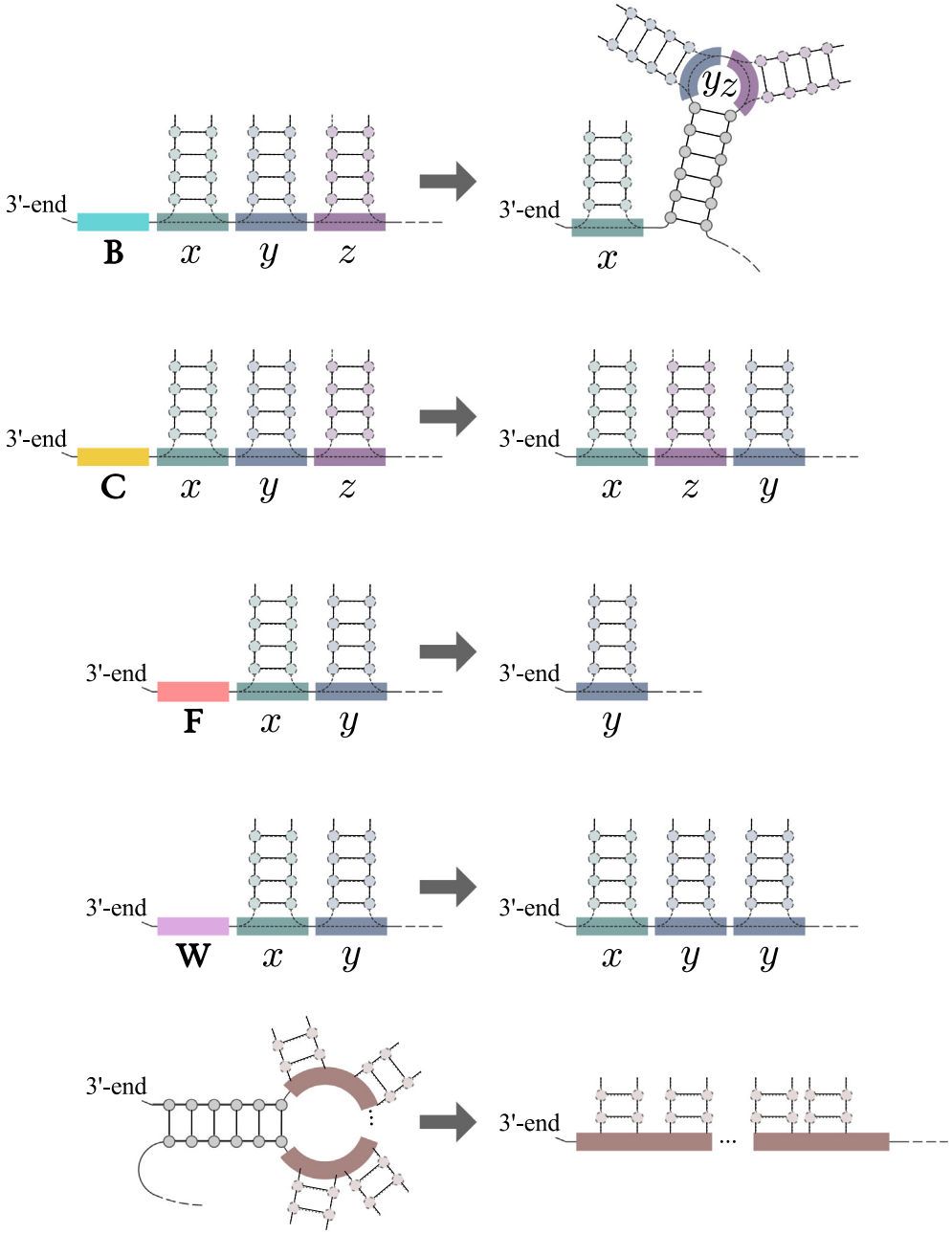

Fig. 4. *Model I* – A set of RNA editing rules that mimic the operational rules of left-associative CL based on combinators **B**, **C**, **F** and **W**. In all the rules, RNA elements x, y, and z must either be single combinator motifs (codes that represent **B**, **C**, **F** or **W**) or stem loop bases. The fifth rule corresponds to parenthesis removal when the left most term is enclosed in parentheses. All edits are designed to be applied on the 3′-end of RNA strands.

right-associative CL (see Fig. 5) where *abcde* is interpreted as *a*(*b*(*c*(*de*))). In right-associative CL, the operation rules can be implemented anywhere along the RNA strand, whereas the rules written for left-associative CL were designed to be implemented at the 3′-end of the strand only. It is sufficient for these editing rules to be implemented by distinct enzymes in an uncoordinated fashion to achieve a universal computation system.

The method of nesting terms through RNA stem loops presents an opportunity to implement addressable memory and variable substitution. One such implementation is illustrated in Fig. 6. In this model, each variable is assigned an *address* (specified by a unique sequence of nucleotides). An RNA strand containing that address sequence at its 5′-end stores the value of that variable. A variable can be referred to by other RNA strands using a *reference* sequence defined as the reverse complementary sequence of the address sequence. Simple cleavage/ligation operations can substitute the reference sequence with the value of the corresponding variable as a nested term as shown in Fig. 6. (For this system to work with the left-associative CL model (Fig. 4), the address and reference sequences must themselves fold into stem loops).

An addressable memory system with variable substitution has many advantages. First it allows longer strands to be broken down to shorter ones while maintaining a link between the strands. This overcomes some of the physical limits on the lengths of RNA programs and parallelizes computation. Second, it allows coordination across many programs that share the same memory space. A program can write the result of a computation in a designated memory address that another program uses as an input. Third, it

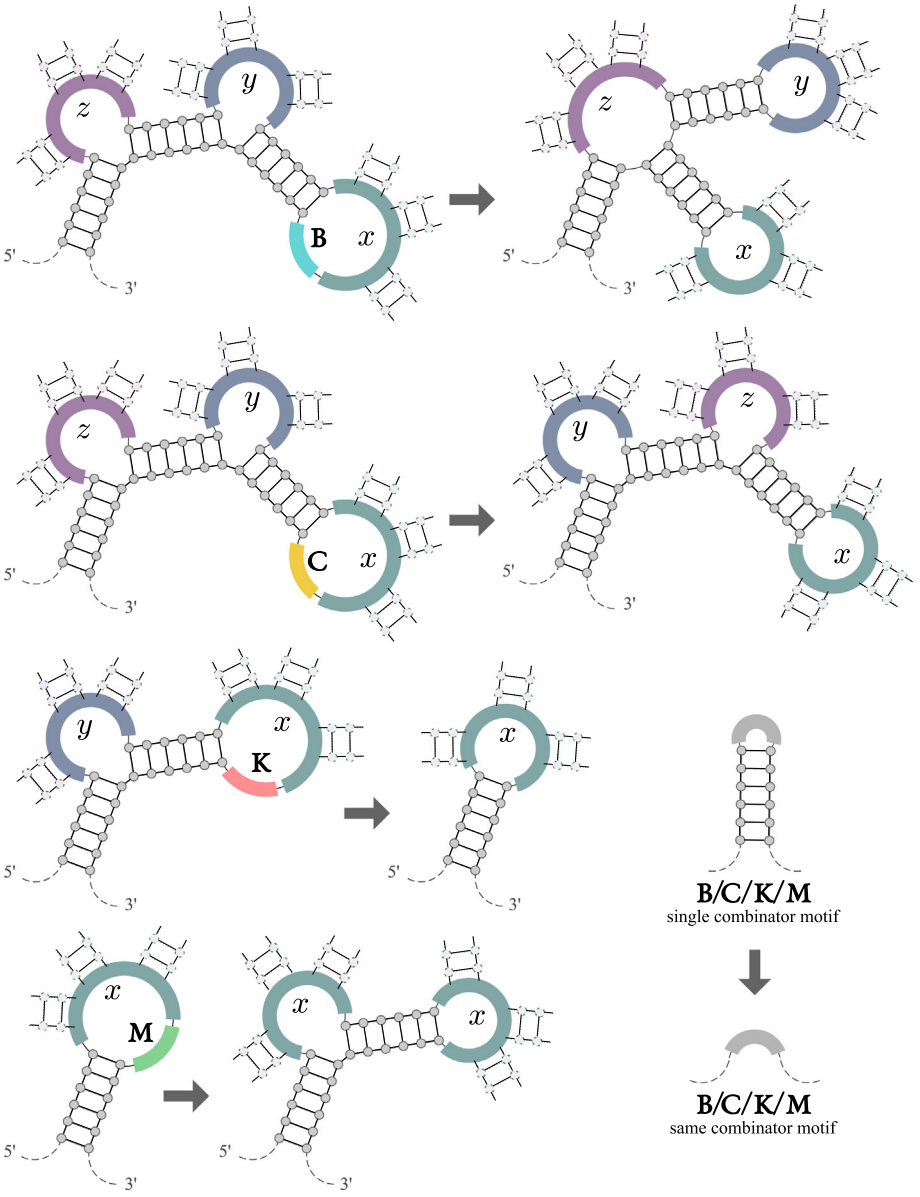

Fig. 5. *Model II* – A set of RNA editing rules that mimic the operational rules of right-associative CL based on combinators **B**, **C**, **F** and **K**. In all the rules, RNA elements x, y, and z can be any non-empty sequence representing primitive or composite combinator terms. The fifth rule corresponds to parenthesis removal when a single combinator is enclosed in parentheses. In contrast to Model I (Fig. 4) editing can happen anywhere along an RNA strand, even multiple locations at once. If multiple locations can be edited, the order of operations does not affect the final result (according to the Church-Rosser theorem).

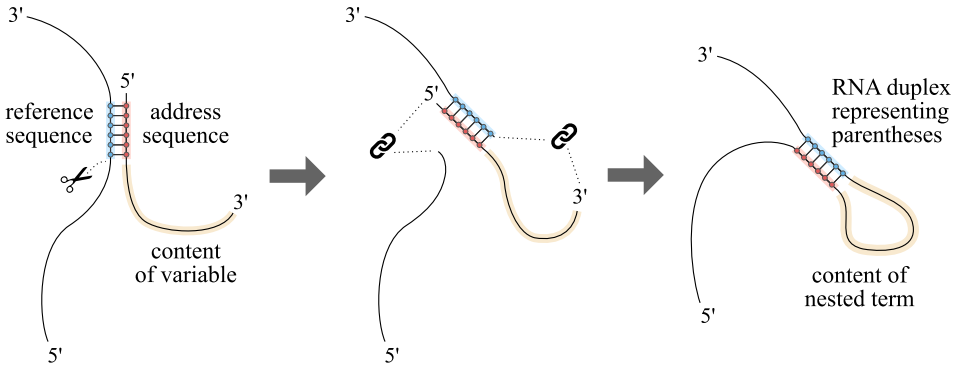

Fig. 6. *Addressable Memory* – An example mechanism that can implement variable substitutions through RNA, assuming stem loops hold nested terms. First two distinct RNA strands come together, one containing the address and content of a variable and the other containing a reference sequence that base-pairs with the address sequence. One cleavage and two ligations as illustrated is sufficient to incorporate the content of the variable as a stem loop at the location it is referenced. This process in reverse can also be used to excise a loop from a stem-loop, while maintaining the link across two strands.

can be more efficient to work with variable representations. For example, if a nested term is to be duplicated only to have one copy deleted, it is more efficient to have it represented by a short reference sequence and duplicate the reference sequence. And fourth, term permutations can be done much quicker using addresses and references. The first 61 steps of the program in Fig. 2B only rearranges the inputs, but can be accomplished with only 5 variable substitution operations.

2.3. Standard DNA-dependent transcription eliminates the need for RNA duplication

Above, we evaluated the plausibility of *deletion*, *reordering*, *duplication*, and *nesting* in RNA molecules. Deletion, reordering, and nesting can be done with O(1) number of cleavage/ligation operations, but duplication requires RNA synthesis and its number of operations is proportional to the length of the element that is being duplicated. Furthermore, RNA replication from RNA templates is not yet known to occur widely across all cells of all lifeforms. In this section I show that RNA directed RNA replication is not strictly required for universal computation. To make this possible I will abandon pure combinatory logic as this model requires an addressing system where self-referencing (or cycles in the reference graph) is permitted.

Strictly speaking, it is not permitted to define a function in terms of itself in λ-calculus or CL. For instance the function ***f* = C (C(BC(CC)))(CK) *f*** must be redefined in terms of the **Y** combinator as ***f* = Y C(C(BC(CC)))(CK) = B(WI)(BWB)C(C(BC(CC)))(CK)** (see Appendix C). (The founders of λ-calculus/CL were concerned with creating a sound foundation for mathematics that avoids contradictions and ill-defined self-referencing terms). But if we relax this constraint it becomes possible to achieve universal computation without any duplicating combinators (such as **W** or **M**). Instead, the three combinators **B**, **C**, and **K** alongside an addressing system is sufficient to simulate any Turing machine with linear slowdown (see Appendix C for a constructive proof that uses B, C, and K, and self-reference to linearly implement any Turing machine). Note that in the model of right-associative CL (Fig. 5) applications of B, C, K, and parenthesis removal can be fully accomplished by merely cleaving and ligating RNA at predetermined positions relative to the combinator motif. Except for the application of the **M** combinator (which is no longer needed for universality), the number of duplexes (i.e., parentheses) on both sides of the rules are the same. Therefore, even duplex insertion is not strictly required.

We can combine the addressing system of Fig. 6 with the right associative model of Fig. 5, excluding the M combinator – which is the combinator that cannot be implemented purely by cleavage and ligation. The resulting model relies on perpetual transcription of RNA strands from static genomic templates. The transcripts can then be recursively inserted into one another guided by an addressing system.

In order to prevent infinite loops of self-insertion where multiple copies of a self-referencing transcript get inserted into one another, the value substitution mechanism of Fig. 6 can be reformulated to only occur when a reference is at the left-most position (the 3′-end of the strand) or when it is enclosed in parentheses on both sides. An alternative solution is to split a reference sequence into two subsequences only to have them joined when value substitution is needed in the algorithm.

2.4. Simulating an example Program: Addition of two arbitrarily large numbers

To better illustrate the idea of computation through RNA, I will explain how to perform arithmetic addition using the RNA model described in section 2.3. Note that the problem of adding two bounded numbers (say two 8 bit numbers, between 0 and 256) can be solved using a static memory-less circuit of boolean logic gates. But our example demonstrates how to use recursion to solve addition with no bounds on the input size, apart from resource constraints like time, space, and energy.

In our example, a numbers is represented as a strand with distinct RNA elements for binary digits 1 and 0. (Examples of the numbers 6 and 2, respectively $(110)_2$ and $(10)_2$ in binary form, are depicted in Fig. 7C). Two RNA strands shown in Fig. 7A & 7B need to be perpetually transcribed from DNA templates. These two transcripts (hereon referred to as transcripts A and B) have unique address sequences at the 5′-end and can be referred to by the reverse complementary sequence of those address sequences. For instance the RNA strand in Fig. 7C has a reference to transcript A near its 3′-end which should lead to transcript A being inserted at that location through the rule illustrated in Fig. 6. Additionally, transcript A has a reference to itself and a reference to transcript B. Transcript B also has a reference to itself.

Here, we only use three combinators: B, C, and K. The rules for applying them are shown in Fig. 5. Except for the rule for applying the M combinator (which we make no use of), all of the editing rules in Fig. 5 can be implemented by only cleavage and ligation at fixed positions relative to the combinator motifs. It is assumed that hypothetical enzymes detect these motifs and collectively implement these rules in any order. To see the full derivation of the addition program, refer to Appendix C section 6.1.

To calculate 6 + 2 = 8, at least 4 copies of transcript A and 5 copies of transcript B need to be present. The final product is an RNA strand that contains four RNA elements corresponding to digits $(1000)_2$ which represents the number 8. The strength of this method becomes evident when examining how the number of operations scales with the size of the input. By simulating the editing rules for inputs up to 10^6 I show that the number of operations scales logarithmically relative to the input (Fig. 7D). This would not have been possible had I used a unary encoding like Church numerals. (See supplementary data for simulation code).

The addition program in Fig. 7 has arbitrary details and conventions (how to represent a number, which primitive combinators to use, etc). I constructed it for illustration purposes only and it is not intended to resemble how arithmetics is done in real biological cells. This example shows a recursive program can be computed with purely cleavage, ligation, and transcription from static templates. A detailed proof (Appendix C section 7) shows that any computable function can be similarly computed using the same framework.

3. Discussion

This paper proposes a class of theoretical RNA-based models that are able to compute any computable function with identical algorithmic complexity to that of Turing machines. There is plenty of flexibility in the details of the models. The choice of primitive combinators is arbitrary and there are an infinite variety of primitives that lead to universal scope (e.g. SKI basis set or the BCKW basis set). There is also some freedom in other aspects of the model, such as the parenthesis convention (Figs. 4 and 5 implement left-associative and right associative, respectively).

In one variation, I use an address/reference system to circumvent the requirement of duplicating RNA elements (which is needed for implementing the M or W combinators). Appendix C contains a proof that this model (which – strictly speaking – deviates from CL and λ-calculus because of its use of unbounded variables) is Turing-equivalent in scope and in algorithmic complexity (runtime and memory). A recursive function *f*, defined in terms of itself, can be executed by self-insertion of an RNA transcript repre-

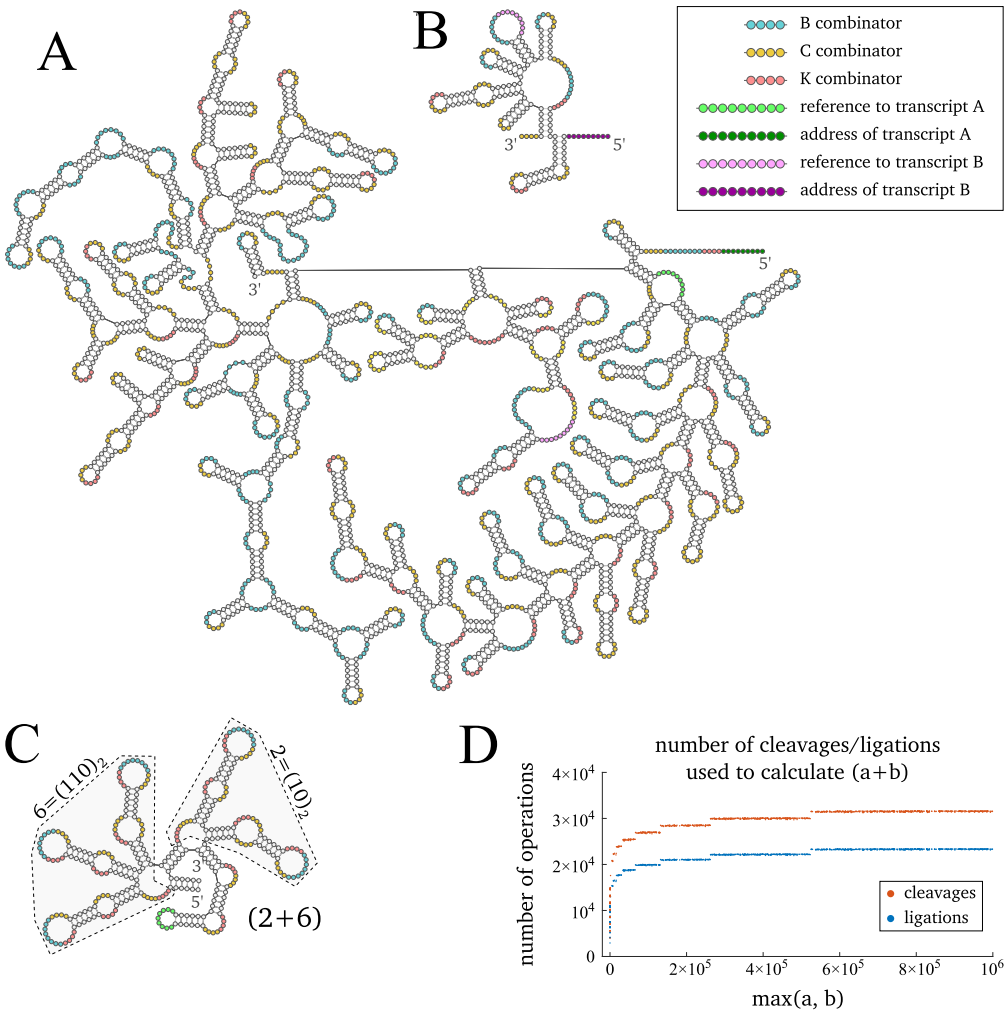

Fig. 7. *Implementation of Addition* – An RNA program was constructed to add arbitrarily large numbers in logarithmic time using the editing rules of Model II (Fig. 5) and an address/reference system similar that of Fig. 6. This program involves two transcripts that are perpetually transcribed from static genomic templates and contain references to themselves. Since the **M** combinator is not used, the rules can be implemented by merely cleaving and ligating RNA strands at predefined positions. Combinator motifs were assumed to be 4-nt long and duplexes are each 6-bp. **(A)** Self-referencing 2518-nt transcript responsible for addition that also contains a reference to transcript B. **(B)** Self-referencing 202-nt transcript responsible for reversing the content of a stack (or equivalently reversing the digits of a number). **(C)** an RNA program that computes 2 + 6 by referencing transcript A. **(D)** Simulation of the operation rules demonstrate that the number of operations increase logarithmically relative to the larger addend. (See Appendix C for derivations of the transcripts A and B).

senting *f*. In the process of computing *f* up to a depth of n, exactly n copies of the transcript are consumed and degraded. In a sense, the RNA-based RNA duplication of the purely CL models can be outsourced to the more plausible DNA-based RNA transcription. We already know the biological processes for perpetual RNA transcription from static genomic templates. In order to validate this model, what is left to discover are the specific enzymatic reactions that detect motifs and carry out the local cleavage/ligation operations at fixed positions relative to those motifs.

The components that are needed to implement these models are so familiar and unsophisticated that it appears plausible that nature may have already achieved an RNA-based universal computation system. In contrast to previous DNA/RNA-based models that achieve Turing-equivalence (Bennett, 1982; Bryant, 2012; Geary et al., 2019; Lakin and Modelling, 2011; Mao et al., 2000; Qian et al., 2011; Rothemund, 1996; Rothemund et al., 2004; Shapiro, 2012; Varghese et al., 2015; Winfree et al., 1998; Woods et al., 2019; Yahiro and Hagiya, 2016; Geary et al., 2017; Shapiro and Karunaratne, 2001), these new models that bridge combinatory logic with RNA molecular biology are an attempt to conceive of a computation system within the limits of what may already exist in cells. This work motivates us to search for nature's universal computer at the molecular level and suggests some guiding details that may aid us in its discovery.

3.1. Challenges to the theory

The formation of stem loops is an indispensable component of the models presented in this paper as it solves the otherwise difficult problem of parenthesis matching. But RNA secondary structure is not necessarily fixed or unique. The same RNA strand can change shape and fold into different kinetically stable structures (Marek et al., 2011). An estimated 20–50% of mRNAs assume alternate conformations (Lu et al., 2016). This can be a problem for a model that depends on stems representing parentheses because it makes RNA sequences ambiguous in their representations of CL terms. Even randomly generated nucleotide sequences lead to complex secondary structures that resemble some of the statistical properties of naturally occurring secondary structure, including non-local duplexes joining regions near the 3′ and 5′ ends (Fang et al., 2018; Yoffe et al., 2011). Additionally, even if an RNA strand starts off in a highly stable state, local edits may radically change

the thermodynamic energy landscape and lead to changes in secondary structure that do not reflect the intended operation rules of CL. Stable and controlled secondary structure is a critical part of the model.

How can the theory be reconciled with the unfettered nature of RNA structure? There are many factors that can influence the stability of secondary structure. Translation has a destabilizing effect on structure in mRNA (Mustoe et al., 2018). So non-protein-coding RNA that are not subject to translation may be more stable than the estimates we have for mRNA. RNA binding proteins or other factors may, in theory, stabilize stem loops as an RNA is being prepared to undergo CL modifications. Also, some sequences form more stable duplexes than others and it may be the case that only highly stable duplexes are treated as parentheses. G-C bonds are stronger than A-U bonds (Mathews et al., 2004) and restrictions on the kinds of motifs that constitute parentheses can disambiguate them from the motifs that represent primitive combinators. And finally, there are many ways to construct the same CL term using primitive combinators. Semantically inconsequential variations to syntactic representation of terms can also be used to prevent the formation of unintended duplexes. For example, any number of identity combinators can be inserted after an open parenthesis without altering the intended computation, and that by itself may stabilize secondary structure. Importantly, none of the above models rely on strand displacement or the unwinding of already formed RNA duplexes. Once a RNA stem is formed, it never needs to come apart until that stem is excised and degraded. This makes RNA binding proteins an excellent candidate for stabilizing duplexes for the lifetime of an RNA strand. Conformational heterogeneity of RNA strands does pose a challenge to these models, but not an unsurmountable one.

Stochasticity may also occur at the level of primary structure. This can be dealt with by error-correction strategies. The most obvious error-correction method is programmatic error-correction, e.g. the use of check-sums or verification of program outputs. Additionally, an RNA based program can spawn many copies of the same sub-program in parallel. Some fraction of the programs may accumulate errors during computation but the overseeing program can select the most frequently occurring result. Combinatory logic is as powerful as any typical programming language and is capable of implementing any of the software-based error-correction strategies adopted in modern computing technology.

Another challenge for a solely RNA based molecular engram theory is RNA stability. If a molecule were to serve as a memory engram it must at least exhibit stability over similar time periods as cognitive memories. RNA molecules have an average half-life of around 7 hrs (Clark et al., 2012; Tani et al., 2012). DNA, however, can last for a lifetime and RNA information can be converted back to DNA through reverse transcription. In recent decades, we have learned that different cells of the same individual very often have differences in their DNA (Biesecker and Spinner, 2013). Neurons exhibit exceptionally high DNA diversity in what is known as neuronal *somatic mosacism* (Cai et al., 2014; McConnell et al., 2013; Upton et al., 2015; Westra et al., 2010). Various forms of DNA modifications exist in neurons, including single nucleotide variations, duplications, copy number variations, rearrangements, and aneuploidy. The quantitative extent of somatic mosaicism in the brain is currently under scrutiny (Evrony, 2016; Treiber and Waddell, 2017) but it is especially pronounced in the human frontal cortex where average DNA content is enlarged by an estimated 4% (Westra et al., 2010) with a substantial fraction of neurons possessing unique massive copy number variations, on the scale of millions of basepairs long (McConnell et al., 2013). An important driver of somatic genomic diversity is *retrotransposition* (Coufal et al., 2009), the process of transcription from DNA to RNA and subsequent *reverse transcription* from RNA back to DNA, resulting in duplications of specific genes. Until recently, it was not known if retrotransposition was restricted to mitotic progenitor cells, i.e., cells that divide to create more cells, or whether it also occurs naturally in post-mitotic cells (Bodea et al., 2018; Rohrback et al., 2018). This is important because reverse transcription is unlikely to facilitate cognitive memory storage unless it is widespread in healthy post-mitotic brain cells. Remarkably, somatic gene recombination, the process that has so far only been found in immune cells, was recently reported in post-mitotic healthy human frontal cortex neurons (Chai and Gleeson, 2018; Lee and Chun, 2019; Lee et al., 2018). The validity of this study is under debate and the results have not yet been independently replicated (Lee et al., 2020; Kim et al., 2020). But an independent group has recently demonstrated retrotransposition in post-mitotic cultured neuronal cells (Macia et al., 2017). These findings support the idea that RNA-based computation system can store the results of computation back into the form of DNA sequences as long-term memory. This idea was previously formulated, leading to the prediction that *"individual neural cell will have distinctive spatially and temporally defined genomic sequences and chromatin structure*" (Mattick and Mehler, 2008). In fact, most long non-coding RNA are localized in the nucleus and chromatin-associated (Guttman et al., 2009; Khalil et al., 2009; Mattick, 2018; Tsai et al., 2010). Beyond neurons and lymphocytes, RNA-mediated genomic editing is known to occur in single-celled eukaryotes (Chen et al., 2014; Nowacki et al., 2008) and prokaryotes (Mohanraju et al., 2016). The long-term and short-term categorization of memory may be a reflection of two forms of RNA-based and DNA-based memory storage (Mattick and Mehler, 2008).

Can RNA modifications occur fast enough to potentially facilitate cognition? Two of the most well-studied RNA processes are transcription and translation. RNA Polymerase II transcribes RNA molecules at a rate of 18–100 nt/s equivalent to 36–200 bits/s (Darzacq et al., 2007; Koš and Tollervey, 2010; Landenmark et al., 2015; Malinen et al., 2012; Milo et al., 2010). And the ribosome translates RNA to protein at a speed of roughly 5–11 aa/s equivalent to 30–66 bits/s (Milo et al., 2010; Guet et al., 2008; Olofsson et al., 1987; Siwiak et al., 2013; Wang et al., 2013). It is difficult to quantify how fast animals think but studies of different languages show that the information rate of human speech is roughly on the order of 40 bits/s (languages that are spoken faster have lower bits per syllable than languages that are spoken slower) (Coupé et al., 2019). These two information rates are within the same order of magnitude and also consistent with the spike rates of typical neurons (up to 100–200 Hz). This means that RNA operations can in principle be fast enough to encode/transmit ideas communicated in speech as single RNA molecules. (See Glaser et al., 2013 for a discussion on the feasibility of recording spiking activity in the form of synthetic polynucleotides from an engineering perspective, rather than an evolutionary one). It remains to be shown that modifications of RNA molecules can occur fast enough to execute computation programs that underly animal cognition. In the models described in this paper, addition of two numbers requires on the order of 10^4 cleavage/ligation operations. We do not know the speed at which these hypothetical editing rules may be implemented. But the spliceosome, which implements two cleavages and one ligation, takes an estimated 30 s to slice out an intron (Hnilicová and Staněk, 2011; Huranová et al., 2010). For an RNA-based computation system to facilitate cognition, either more efficient RNA programs must exist or RNA modifications must occur at extremely rapid rates.

3.2. Conclusion

I propose the theory that an RNA-based computation system exists in living cells that is Turing-equivalent, i.e., capable of computing any computable function. This is significant because life is confronted with problems of computation almost everywhere we look, from animal cognition, to single-cell decision-making and embryonic development. We have internalized the adequacy and the generality of universal computation in technology as we equip almost every device with a microprocessor. Likewise, an RNA-based universal computation system may be used to solve the many seemingly dissimilar problems of biology.

The specific models outlined in this paper are meant as a proof of principle that universal computation is well-within the reach of molecular biology without needing to invoke implausible molecular processes. The models implement combinatory logic through RNA molecules. Algorithmic complexities are identical to that of Turing machines (e.g., see Fig. 7D). Parentheses are represented by reverse complementary sequences and RNA secondary structure elegantly solves the problem of parenthesis matching which is crucial to the operations of CL and λ-calculus. Universal computation is achieved through a decentralized process where distinct memory-less enzymes make local modifications to RNA molecules according to basic rules. The modifications can happen in parallel across many RNA strands – even at multiple locations within the same strand – and the order of operations do not matter. The system does not require a central processing unit or anything resembling a Turing machine tape head. No component of the model approaches the complexity of the ribosome. Instead, computation is carried out collectively by uncoordinated enzymes, which may themselves be catalytic or self-cleaving/self-ligating RNAs.

The fact that it is easy to imagine a universal computation system using RNA, while it is difficult to do so using network models, is highly suggestive that life's universal computation system – if there is one – resides in the subcellular domain and involves polynucleotides. An RNA-based computation system may have evolved in the very early stages of life – possibly before the evolution of DNA and proteins, consistent with the RNA-World theory (Gilbert, 1986; Higgs and Lehman, 2015). (The third model depends on DNA-based transcription. But the first two models (Figs. 4 & 5) do not necessarily depend on DNA or protein. Instead, they rely on RNA-dependent RNA replication, which is thought to have existed in the hypothetical RNA World). It is quite easy to stumble upon universal computation in systems that use a symbolic substrate that can grow in its number of symbols (Rendell and Adamatzky, 2002; Cook, 2004; Cardone and Hindley, 2006; Schönfinkel, 1924). Once a universal computation system is established it is hard to see why it would be discarded throughout evolution. It is possibly the case that the RNA-based computation system is the very engine of evolution (Mattick, 2009), optimizing mutations of offspring, consistent with recent evidence surrounding intergenerational inheritance of acquired traits (Benito et al., 2018; Wang et al., 2017).

Two major evolutionary events can be reinterpreted in the context of this theory. First, the evolution of complex multicellular organisms – which requires intracellular communication and sophisticated schemes of cooperation and division of labor (Brunet and King, 2017; Cavalier-Smith, 2017; Knoll, 2011; Niklas and Newman, 2013) – potentially involved a general method of coordination across the RNA-based computation systems of different cells of the same somatic lineage. If the language of computation is encoded as RNA, it is reasonable to consider the theory that the messages conveyed across these cells are primarily in the form of RNA molecules. This is consistent with recent evidence on extracellular trafficking of RNA (Ashley et al., 2018; Bär et al., 2019; Bayraktar et al., 2017; Dinger et al., 2008; Pastuzyn et al., 2018). Second, the evolution of neurons and brains – which is thought to have occurred in independent lineages in metazoans (Burkhardt and Sprecher, 2017; Moran et al., 2015; Moroz, 2009; Moroz and Kohn, 2016) – may have served the purpose of rapid communication across cells, rather than serving as an entirely parallel computation system. Electrical/ionic signaling permits fast information transfer and coordinated motility in large multicellular organisms and has even evolved in plants for fast movement and decision-making based on information collected from sources that are many millimeters apart (Böhm et al., 2016; Choi et al., 2016; Fromm and Lautner, 2007; Hedrich and Neher, 2018). If prior to neurons the language of intracellular signaling was in the form of RNA, it is reasonable to consider the theory that neural signals encode RNA sequences that would have otherwise taken too long to export to downstream target cells. In this view, computation is primarily mediated through subcellular RNA-based processes, and the results of computations are then synaptically transmitted between cells to be used in other computations. This is consistent with the idea that the memory engram is in the form of polynucleotides – not synaptic plasticity (Gallistel and King, 2009; Gallistel, 2017). Of course, this is also consistent with the existence of purely network based computations that do not directly involve RNA; network models can augment an RNA-based computation system.

The idea that polynucleotides are the substrate of memory can be traced back to the 1960s, when the discovery of DNA inspired a new scientific approach rooted in the hypothesis that the memory engram is a macromolecule (Eigen and de Maeyer, 1966; Gaito, 1961; Gaito, 1963; Hechter and Halkerston, 1964; Hydén, 1961; Behavior et al., 1967; Landauer, 1964; Schmitt, 1962; Schmitt, 1966; Schmitt, 1967). It was hypothesized that "every idea is represented uniquely by a macromolecule with particular composition and sequence of monomer constituents" (Schmitt, 1967) and that "learning and memory depend on changes in genic material (or the by-products of genic activity) either in the nucleus or the cytoplasm of the nerve cell soma" (Gaito, 1961), and that electrical signals in the nervous system are converted into nucleotide sequences through a hypothetical RNA transduction mechanism (van Sickle and McCluer, 1966). It was even suggested at the time that the molecular processes that underlie learning and encode new memories may be "*continuous across the phyla (as genetic codes are) and therefore would be reasonably similar for a protozoan and a mammal*" (Gershman et al., 2021; Gelber, 1962). These ideas were largely abandoned in the 1970s (Gaito, 1976; Glassman, 1969; Ungar, 1973; Uphouse et al., 1974) but have been rekindled in recent years (Gallistel, 2017; Langille and Gallistel, 2020; Mattick and Mehler, 2008; Abraham et al., 2019; Gallistel, 2018; Gallistel and Balsam, 2014; Queenan et al., 2017). The recent revival is rooted in the sobering realization that current theories of synaptic plasticity and network activity cannot explain learning, memory, and cognition (Gallistel and King, 2009; Gallistel and Matzel, 2013) and that several lines of evidence bring into question the theory that synaptic strengthening/weakening is the primary form of long-term information storage in the brain (Gallistel, 2017; Cai et al., 2012; Daou and Margoliash, 2020; Jirenhed et al., 2017; Johansson et al., 2014; Johansson et al., 2015; Pearce et al., 2017; Poo et al., 2016; Ryan et al., 2015; Santin and Schulz, 2019; Zhao et al., 2019).

Recently, combinatory logic was proposed as an example of an "assembly language for cognition" (Piantadosi, 2021). While there are a number of challenges in explaining cognition using an RNA-based model of CL, this theory is well-posed to resolve some of the challenges that current network models of cognition face in neuroscience. (See Langille and Gallistel, 2020 for a comparison of the challenges of the connectionist/associative theories of memory and computational/representational theories of memory). One of the major criticisms of symbolic computation models of cognition is that they imply that computation is sequential and single-threaded,

rather than parallel and distributed (Rumelhart and McClelland, 1986). But the models I have presented are unique in that while easily allowing recursion and nested structures, they are also highly parallel. Thanks to the Church-Rosser theorem, a program can be split into numerous RNA strands, or even distributed along of the same RNA strand to be computed in parallel. In principle, an RNA implementation of CL allows for unbounded parallelization with no theoretical limit to the number of threads; modern-day computers, on the other hand, are limited by the number of their CPU cores.

The theory of natural RNA-based universal computation makes the falsifiable prediction that RNA molecules are modified in ways that radically deviate from their genomic templates and that these modifications are causally involved in cognition, cell-behavior, and/or development. Lengthening of RNA strands, through either duplication of RNA elements or integration of one strand into another, appears to be inevitable for fulfilling *memory expansion* (which is a necessary condition for any universal computation system in which memory usage is well-defined). We currently have very limited evidence for the existence of endogenous RNA sequences that cannot be mapped to the genome or accounting for through known RNA processing mechanisms. Some examples include widespread single-nucleotide variations (that cannot be attributed to ADAR or APOBEC enzymes) (Li et al., 2011; Wang et al., 2014), non-genomically encoded 5′-poly(U) tails (Kapranov et al., 2010), exon repetition (Dixon et al., 2007; Frantz et al., 1999; Rigatti et al., 2004), post-transcriptional exon shuffling (Al-Balool et al., 2011; Dixon et al., 2005), and chimeric transcripts resulting from post-transcriptional fusion of RNA molecules (Goymer, 2008; Li et al., 2008; Singh et al., 2020; Tang et al., 2019), although some of these results have been controversial (Li et al., 2012; Kleinman and Majewski, 2012; Lin et al., 2012; Pickrell et al., 2012). The scarcity of evidence for extensive RNA editing is not definitive; detection of RNA sequences that are cell-specific or expressed in low numbers is notoriously difficult with current technology (Gloss and Dinger, 2016). RNA sequence analysis pipelines typically discard reads that cannot be mapped to the genome or, in the case of *de novo* transcript assembly, discard aberrant reads that do not match other reads. It may be possible to discover the editing motifs of the hypothetical RNA-based system by careful examination of RNA sequencing data where unexpected editing patterns occur at low frequency. This is perhaps the simplest and most obvious first step for empirically testing the theory. To ultimately validate the theory, one needs to show that specific non-protein-coding RNA species are causally necessary for cognition and/or development, such that novel modifications to their sequences will have predictable effects in animal behavior and/or ontogeny. Conversely, the theory can be falsified if it is shown that the non-protein-coding content of the genome is mostly dispensable or can be replaced with random sequences without completely disrupting cognition, development, and cell function. Even if the RNA-based theory is disproved, the fact that universal computation is molecularly feasible and within the reach of evolution suggests that we must then search for life's universal computation system elsewhere.

CRediT authorship contribution statement

Hessameddin Akhlaghpour: Conceptualization, Methodology, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

I would like to thank Gaby Maimon (Rockefeller University), Jeremy Dittman (Weill Cornell Medicine), Abbas Rizvi (Columbia University), John Mattick (University of New South Wales), and Charles Randy Gallistel (Rutgers University) for the discussions and their insightful comments on this work.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jtbi.2021.110984.

References

Abraham, W.C., Jones, O.D., Glanzman, D.L., 2019. Is plasticity of synapses the mechanism of long-term memory storage? npj Sci. Learn. 4, 9. https://doi.org/10.1038/s41539-019-0048-y.

Adleman, L.M., 1994. Molecular computation of solutions to combinatorial problems. Science 266 (5187), 1021–1024.

Al-Balool, H.H., Weber, D., Liu, Y., Wade, M., Guleria, K., Nam, P.L.P., Clayton, J., Rowe, W., Coxhead, J., Irving, J., Elliott, D.J., Hall, A.G., Santibanez-Koref, M., Jackson, M.S., 2011. Post-transcriptional exon shuffling events in humans can be evolutionarily conserved and abundant. Genome Res. 21 (11), 1788–1799. https://doi.org/10.1101/gr.116442.110.

Aprea, J., Calegari, F., 2015. Long non-coding RNAs in corticogenesis: deciphering the non-coding code of the brain. EMBO J. 34, 2865–2884. https://doi.org/10.15252/embj.201592655.

Asarin, E., Maler, O., Pnueli, A., 1995. Reachability analysis of dynamical systems having piecewise-constant derivatives. Theor Comput Sci. 138 (1), 35–65. https://doi.org/10.1016/0304-3975(94)00228-B.

Ashley, J., Cordy, B., Lucia, D., Fradkin, L.G., Budnik, V., Thomson, T., 2018. Retrovirus-like gag protein arc1 binds RNA and traffics across synaptic boutons. Cell 172 (1-2), 262–274.e11. https://doi.org/10.1016/j.cell.2017.12.022.

Baluška, F., Levin, M., 2016. On having no head: cognition throughout biological systems. Front. Psychol. 7. https://doi.org/10.3389/fpsyg.2016.00902.

Bär, C., Thum, T., de Gonzalo-Calvo, D., 2019. Circulating miRNAs as mediators in cell-to-cell communication. Epigenomics. 11 (2), 111–113. https://doi.org/10.2217/epi-2018-0183.

Bartonicek, N., Clark, M.B., Quek, X.C., Torpy, J.R., Pritchard, A.L., Maag, J.L.V., Gloss, B.S., Crawford, J., Taft, R.J., Hayward, N.K., Montgomery, G.W., Mattick, J.S., Mercer, T.R., Dinger, M.E., 2017. Intergenic disease-associated regions are abundant in novel transcripts. Genome Biol. 18 (1). https://doi.org/10.1186/s13059-017-1363-3.

Bayraktar, R., Van Roosbroeck, K., Calin, G.A., 2017. Cell-to-cell communication: microRNAs as hormones. Mol. Oncol. 11 (12), 1673–1686. https://doi.org/10.1002/1878-0261.12144.

Behavior, H.H., 1967. Neural function, and RNA. In: Davidson, J.N., Cohn, W.E. (Eds.), Progress in Nucleic Acid Research and Molecular Biology. Academic Press, pp. 187–218. https://doi.org/10.1016/S0079-6603(08)60527-9.

Benenson, Y., 2009. RNA-based computation in live cells. Curr. Opin. Biotechnol. 20 (4), 471–478. https://doi.org/10.1016/j.copbio.2009.08.002.

Benenson, Y., 2009. Biocomputers: from test tubes to live cells. Mol. BioSyst. 5, 675–685. https://doi.org/10.1039/B902484K.

Benenson, Y., 2012. Biomolecular computing systems: principles, progress and potential. Nat. Rev. Genet. 13 (7), 455–468. https://doi.org/10.1038/nrg3197.

Benito, E., Kerimoglu, C., Ramachandran, B., Pena-Centeno, T., Jain, G., Stilling, R.M., Islam, M.R., Capece, V., Zhou, Q., Edbauer, D., Dean, C., Fischer, A., 2018. RNA-dependent intergenerational inheritance of enhanced synaptic plasticity after environmental enrichment. Cell Rep. 23 (2), 546–554. https://doi.org/10.1016/j.celrep.2018.03.059.

Bennett, C.H., 1973. Logical reversibility of computation. IBM J. Res. Dev. 17 (6), 525–532. https://doi.org/10.1147/rd.176.0525.

Bennett, C.H., 1982. The thermodynamics of computation—a review. Int. J. Theor. Phys. 21 (12), 905–940. https://doi.org/10.1007/BF02084158.

Biebricher, C.K., Luce, R., 1996. Template-free generation of RNA species that replicate with bacteriophage T7 RNA polymerase. EMBO J. 15 (13), 3458–3465. https://doi.org/10.1002/j.1460-2075.1996.tb00712.x.

Biebricher, C.K., Orgel, L.E., 1973. An RNA that multiplies indefinitely with DNA-dependent RNA polymerase: selection from a random copolymer. Proc. Natl. Acad. Sci. 70 (3), 934–938. https://doi.org/10.1073/pnas.70.3.934.

Biesecker, L.G., Spinner, N.B., 2013. A genomic view of mosaicism and human disease. Nat. Rev. Genet. 14 (5), 307–320. https://doi.org/10.1038/nrg3424.

Bodea, G.O., McKelvey, E.G.Z., Faulkner, G.J., 2018. Retrotransposon-induced mosaicism in the neural genome. Open Biol. 8 (7), 180074. https://doi.org/10.1098/rsob.180074.

Böhm, J., Scherzer, S., Krol, E., Kreuzer, I., von Meyer, K., Lorey, C., Mueller, T., Shabala, L., Monte, I., Solano, R., Al-Rasheid, K.S., Rennenberg, H., Shabala, S., Neher, E., Hedrich, R., 2016. The venus flytrap dionaea muscipula counts prey-

induced action potentials to induce sodium uptake. Curr. Biol. 26 (3), 286–295. https://doi.org/10.1016/j.cub.2015.11.057.
Boker, U., Dershowitz, N., 2005. How to Compare the Power of Computational Models. pp. 54–64. doi:10.1007/11494645_7.
Boker, U., Dershowitz, N., 2006. Comparing computational power. Log. J. IGPL. 14 (5), 633–647. https://doi.org/10.1093/jigpal/jzl003.
Boker, U., Dershowitz, N., 2008. The church-turing thesis over arbitrary domains. In: Avron, A., Dershowitz, N., Rabinovich, A. (Eds.), Lecture Notes in Computer SciencePillars of Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 199–229. https://doi.org/10.1007/978-3-540-78127-1_12.
Branicky, M.S., 1995. Universal computation and other capabilities of hybrid and continuous dynamical systems. Theor Comput Sci. 138 (1), 67–100. https://doi.org/10.1016/0304-3975(94)00147-B.
Brunet, T., King, N., 2017. The origin of animal multicellularity and cell differentiation. Dev. Cell 43 (2), 124–140. https://doi.org/10.1016/j.devcel.2017.09.016.
Bryant, Barbara, 2012. Chromatin Computation. PLOS ONE 7 (5). https://doi.org/10.1371/journal.pone.0035703.
Burkhardt, P., Sprecher, S.G., 2017. Evolutionary origin of synapses and neurons – bridging the gap. BioEssays 39 (10), 1700024. https://doi.org/10.1002/bies.201700024.
Cabessa, J., Song, T., 2019. Turing complete neural computation based on synaptic plasticity. PLoS ONE 14 (10), e0223451. https://doi.org/10.1371/journal.pone.0223451.
Cabili, M.N., Trapnell, C., Goff, L., Koziol, M., Tazon-Vega, B., Regev, A., Rinn, J.L., 2011. Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. Genes Dev. 25 (18), 1915–1927. https://doi.org/10.1101/gad.17446611.
Cai, X., Evrony, G., Lehmann, H., Elhosary, P., Mehta, B., Poduri, A., Walsh, C., 2014. Single-cell, genome-wide sequencing identifies clonal somatic copy-number variation in the human brain. Cell Rep. 8 (5), 1280–1289. https://doi.org/10.1016/j.celrep.2014.07.043.
Cai, D., Pearce, K., Chen, S., Glanzman, D., 2012. Reconsolidation of long-term memory in aplysia. Curr. Biol. 22 (19), 1783–1788. https://doi.org/10.1016/j.cub.2012.07.038.
Cardelli, L., 2011. Strand algebras for DNA computing. Nat. Comput. 10 (1), 407–428. https://doi.org/10.1007/s11047-010-9236-7.
Cardone, F., Hindley, J.R., 2006. History of lambda-calculus and combinatory logic. Handb Hist Log. 5, 723–817.
Cavalier-Smith, T., 2017. Origin of animal multicellularity: precursors, causes, consequences—the choanoflagellate/sponge transition, neurogenesis and the Cambrian explosion. Philos Trans R Soc B Biol Sci. 372 (1713), 20150476. https://doi.org/10.1098/rstb.2015.0476.
Ceze, L., Nivala, J., Strauss, K., 2019. Molecular digital data storage using DNA. Nat. Rev. Genet. 20 (8), 456–466. https://doi.org/10.1038/s41576-019-0125-3.
Chai, G., Gleeson, J.G., 2018. A newly discovered mechanism driving neuronal mutations in Alzheimer's disease. Nature 563 (7733), 631–632. https://doi.org/10.1038/d41586-018-07334-9.
Chang W-S, Pettersson JH-O, Le Lay C, Shi M, Lo N, Wille M, et al., 2019. Novel hepatitis D-like agents in vertebrates and invertebrates. Virus Evol., 2019;5. doi:10.1093/ve/vez021
Chang, J., Nie, X., Chang, H.E., Han, Z., Taylor, J., 2008. Transcription of hepatitis delta virus RNA by RNA polymerase II. J. Virol. 82 (3), 1118–1127. https://doi.org/10.1128/JVI.01758-07.
Chen, X., Bracht, J., Goldman, A., Dolzhenko, E., Clay, D., Swart, E., Perlman, D., Doak, T., Stuart, A., Amemiya, C., Sebra, R., Landweber, L., 2014. The architecture of a scrambled genome reveals massive levels of genomic rearrangement during development. Cell 158 (5), 1187–1198. https://doi.org/10.1016/j.cell.2014.07.034.
Chen, J., Wood, D.H., 2000. Computation with biomolecules. Proc. Natl. Acad. Sci. 97 (4), 1328–1330. https://doi.org/10.1073/pnas.97.4.1328.
Cheng, J., Kapranov, P., Drenkow, J., Dike, S., Brubaker, S., Patel, S., Long, J., Stern, D., Tammana, H., Helt, G., Sementchenko, V., Piccolboni, A., Bekiranov, S., Bailey, D. K., Ganesh, M., Ghosh, S., Bell, I., Gerhard, D.S., Gingeras, T.R., 2005. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. Science 308 (5725), 1149–1154.
Choi, W.-G., Hilleary, R., Swanson, S.J., Kim, S.-H., Gilroy, S., 2016. Rapid, long-distance electrical and calcium signaling in plants. Annu. Rev. Plant Biol. 67 (1), 287–307. https://doi.org/10.1146/annurev-arplant-043015-112130.
Church, A., 1936. An unsolvable problem of elementary number theory. Am J Math. 58, 345–363. https://doi.org/10.2307/2371045.
Clark, M.B., Amaral, P.P., Schlesinger, F.J., Dinger, M.E., Taft, R.J., Rinn, J.L., Ponting, C. P., Stadler, P.F., Morris, K.V., Morillon, A., Rozowsky, J.S., Gerstein, M.B., Wahlestedt, C., Hayashizaki, Y., Carninci, P., Gingeras, T.R., Mattick, J.S., Eisen, M.B., 2011. The reality of pervasive transcription. PLoS Biol. 9 (7), e1000625. https://doi.org/10.1371/journal.pbio.1000625.
Clark, M.B., Johnston, R.L., Inostroza-Ponta, M., Fox, A.H., Fortini, E., Moscato, P., Dinger, M.E., Mattick, J.S., 2012. Genome-wide analysis of long noncoding RNA stability. Genome Res. 22 (5), 885–898. https://doi.org/10.1101/gr.131037.111.
Cocquerelle, C., Daubersies, P., Majérus, M.A., Kerckaert, J.P., Bailleul, B., 1992. Splicing with inverted order of exons occurs proximal to large introns. EMBO J. 11 (3), 1095–1098.
Cook, M., 2004. Universality in elementary cellular automata. Complex Syst. 40.
Coufal, N.G., Garcia-Perez, J.L., Peng, G.E., Yeo, G.W., Mu, Y., Lovci, M.T., Morell, M., O'Shea, K.S., Moran, J.V., Gage, F.H., 2009. L1 retrotransposition in human neural progenitor cells. Nature 460 (7259), 1127–1131. https://doi.org/10.1038/nature08248.
Coupé, C., Oh, Y.M., Dediu, D., Pellegrino, F., 2019. Different languages, similar encoding efficiency: comparable information rates across the human communicative niche. Sci. Adv. 5, eaaw2594. https://doi.org/10.1126/sciadv.aaw2594.
Daou, A., Margoliash, D., 2020. Intrinsic neuronal properties represent song and error in zebra finch vocal learning. Nat. Commun. 11, 1–17. https://doi.org/10.1038/s41467-020-14738-7.
Darty, K., Denise, A., Ponty, Y., 2009. VARNA: interactive drawing and editing of the RNA secondary structure. Bioinformatics 25, 1974–1975. https://doi.org/10.1093/bioinformatics/btp250.
Darzacq, X., Shav-Tal, Y., de Turris, V., Brody, Y., Shenoy, S.M., Phair, R.D., Singer, R. H., 2007. In vivo dynamics of RNA polymerase II transcription. Nat. Struct. Mol. Biol. 14 (9), 796–806. https://doi.org/10.1038/nsmb1280.
Derrien, T., Johnson, R., Bussotti, G., Tanzer, A., Djebali, S., Tilgner, H., Guernec, G., Martin, D., Merkel, A., Knowles, D.G., Lagarde, J., Veeravalli, L., Ruan, X., Ruan, Y., Lassmann, T., Carninci, P., Brown, J.B., Lipovich, L., Gonzalez, J.M., Thomas, M., Davis, C.A., Shiekhattar, R., Gingeras, T.R., Hubbard, T.J., Notredame, C., Harrow, J., Guigó, R., 2012. The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. Genome Res. 22 (9), 1775–1789. https://doi.org/10.1101/gr.132159.111.
Dershowitz, N., Gurevich, Y., 2008. A natural axiomatization of computability and proof of Church's Thesis. Bull Symb Log. 14 (3), 299–350. https://doi.org/10.2178/bsl/1231081370.
Dexter, J.P., Prabakaran, S., Gunawardena, J., 2019. A complex hierarchy of avoidance behaviors in a single-cell eukaryote. Curr. Biol. 29 (24), 4323–4329. e2. https://doi.org/10.1016/j.cub.2019.10.059.
Dinger, M.E., Mercer, T.R., Mattick, J.S., 2008. RNAs as extracellular signaling molecules. J. Mol. Endocrinol. 40, 151–159. https://doi.org/10.1677/JME-07-0160.
Dinger, M.E., Amaral, P.P., Mercer, T.R., Mattick, J.S., 2009. Pervasive transcription of the eukaryotic genome: functional indices and conceptual implications. Brief Funct. Genomics 8 (6), 407–423. https://doi.org/10.1093/bfgp/elp038.
Dixon, R.J., Eperon, I.C., Hall, L., Samani, N.J., 2005. A genome-wide survey demonstrates widespread non-linear mRNA in expressed sequences from multiple species. Nucleic Acids Res. 33, 5904–5913. https://doi.org/10.1093/nar/gki893.
Dixon, R.J., Eperon, I.C., Samani, N.J., 2007. Complementary intron sequence motifs associated with human exon repetition: a role for intragenic, inter-transcript interactions in gene expression. Bioinformatics 23, 150–155. https://doi.org/10.1093/bioinformatics/btl575.
Doolittle, W.F., 2013. Is junk DNA bunk? a critique of ENCODE. Proc. Natl. Acad. Sci. 110 (14), 5294–5300. https://doi.org/10.1073/pnas.1221376110.
Eigen, M., de Maeyer, L., 1966. Chemical means of information storage and readout in biological systems. Naturwissenschaften 53 (3), 50–57. https://doi.org/10.1007/BF00594747.
Elkon, R., Ugalde, A.P., Agami, R., 2013. Alternative cleavage and polyadenylation: extent, regulation and function. Nat. Rev. Genet. 14 (7), 496–506. https://doi.org/10.1038/nrg3482.
ENCODE Project Consortium, 2012. An integrated encyclopedia of DNA elements in the human genome. Nature 489, 57–74. https://doi.org/10.1038/nature11247.
ENCODE Project Consortium, Birney, E., Stamatoyannopoulos, J.A., Dutta, A., Guigó, R., Gingeras, T.R., et al., 2007. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature 447, 799–816. https://doi.org/10.1038/nature05874.
Ermentrout, G.B., Edelstein-Keshet, L., 1993. Cellular automata approaches to biological modeling. J. Theor. Biol. 160 (1), 97–133. https://doi.org/10.1006/jtbi.1993.1007.
Evrony, G.D., 2016. One brain, many genomes. Science 354 (6312), 557–558.
Fang S, Zhang L, Guo J, Niu Y, Wu Y, Li H, et al. NONCODEV5: a comprehensive annotation database for long non-coding RNAs. Nucleic Acids Res. 2018;46: D308–D314. doi:10.1093/nar/gkx1107.
Fages, François, Le Guludec, Guillaume, Bournez, Olivier, Pouly, Amaury, 2017. Strong turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs. International conference on computational methods in systems biology 10545, 108–127. https://doi.org/10.1007/978-3-319-67471-1_7.
Faulhammer, D., Cukras, A.R., Lipton, R.J., Landweber, L.F., 2000. Molecular computation: RNA solutions to chess problems. Proc. Natl. Acad. Sci. 97 (4), 1385–1389. https://doi.org/10.1073/pnas.97.4.1385.
Flavell, S.W., Kim, T.-K., Gray, J.M., Harmin, D.A., Hemberg, M., Hong, E.J., Markenscoff-Papadimitriou, E., Bear, D.M., Greenberg, M.E., 2008. Genome-wide analysis of MEF2 transcriptional program reveals synaptic target genes and neuronal activity-dependent polyadenylation site selection. Neuron 60 (6), 1022–1038. https://doi.org/10.1016/j.neuron.2008.11.029.
Flores, R., Grubb, D., Elleuch, A., Nohales, M.-Á., Delgado, S., Gago, S., 2011. Rolling-circle replication of viroids, viroid-like satellite RNAs and hepatitis delta virus: variations on a theme. RNA Biol. 8 (2), 200–206. https://doi.org/10.4161/rna.8.2.14238.
Frantz, S.A., Thiara, A.S., Lodwick, D., Ng, L.L., Eperon, I.C., Samani, N.J., 1999. Exon repetition in mRNA. Proc. Natl. Acad. Sci. 96 (10), 5400–5405. https://doi.org/10.1073/pnas.96.10.5400.
Freedman, M.L., Monteiro, A.N.A., Gayther, S.A., Coetzee, G.A., Risch, A., Plass, C., Casey, G., De Biasi, M., Carlson, C., Duggan, D., James, M., Liu, P., Tichelaar, J.W.,

Vikis, You, M., Mills, I.G., 2011. Principles for the post-GWAS functional characterization of cancer risk loci. Nat. Genet. 43 (6), 513–518. https://doi.org/10.1038/ng.840.
Fromm, J., Lautner, S., 2007. Electrical signals and their physiological significance in plants. Plant, Cell Environ. 30, 249–257. https://doi.org/10.1111/j.1365-3040.2006.01614.x.
Gaito, J., 1961. A biochemical approach to learning and memory. Psychol. Rev. 68 (4), 288–292. https://doi.org/10.1037/h0042647.
Gaito, J., 1963. DNA and RNA as memory molecules. Psychol. Rev. 70 (5), 471–480. https://doi.org/10.1037/h0042142.
Gaito, J., 1976. Molecular psychobiology of memory: its appearance, contributions, and decline. Physiol Psychol. 4 (4), 476–484. https://doi.org/10.3758/BF03326600.
Gallistel, C.R., 2017. The coding question. Trends Cogn Sci. 21 (7), 498–508. https://doi.org/10.1016/j.tics.2017.04.012.
Gallistel, C.R., 2018. Finding numbers in the brain. Philos. Trans. R Soc. B Biol. Sci. 373 (1740), 20170119. https://doi.org/10.1098/rstb.2017.0119.
Gallistel, C.R., Balsam, P.D., 2014. Time to rethink the neural mechanisms of learning and memory. Neurobiol. Learn. Mem. 108, 136–144. https://doi.org/10.1016/j.nlm.2013.11.019.
Gallistel, C., King, A., 2009. Memory and the computational brain: why cognitive science will transform. Neuroscience, 288–298. https://doi.org/10.1002/9781444310498.refs.
Gallistel, C.R., Matzel, L.D., 2013. The neuroscience of learning: beyond the Hebbian synapse. Annu. Rev. Psychol. 64 (1), 169–200. https://doi.org/10.1146/annurev-psych-113011-143807.
Geary, C., Meunier, P.-É., Schabanel, N., Seki, S., 2017. Proving the Turing Universality of Oritatami Co-Transcriptional Folding. 2017. Available: https://hal.archives-ouvertes.fr/hal-01567227.
Geary, C., Meunier, P.-É., Schabanel, N., Seki, S., 2019. A computational model for molecular co-transcriptional folding. Int. J. Mol. Sci. 20 (9), 2259. https://doi.org/10.3390/ijms20092259.
Gelber, B., 1962. Acquisition in Paramecium Aurelia during spaced training. Psychol Rec. 12 (2), 165–177. https://doi.org/10.1007/BF03393454.
Gershman, S.J., Balbi, P.E., Gallistel, C.R., 2021. Gunawardena J., Reconsidering the evidence for learning in single cells. Colgin LL, editor. eLife. 10: e61907. doi:10.7554/eLife.61907.
Gilbert, W., 1986. Origin of life: the RNA world 618-618 Nature 319 (6055). https://doi.org/10.1038/319618a0.
Glaser, J.I., Zamft, B.M., Marblestone, A.H., Moffitt, J.R., Tyo, K., Boyden, E.S., Church, G., Kording, K.P., Markel, S., 2013. Statistical analysis of molecular signal recording. PLoS Comput. Biol. 9 (7), e1003145. https://doi.org/10.1371/journal.pcbi.1003145.
Glassman, E., 1969. The biochemistry of learning: an evaluation of the role of RNA and protein. Annu. Rev. Biochem. 38 (1), 605–646. https://doi.org/10.1146/annurev.bi.38.070169.003133.
Gloss, B.S., Dinger, M.E., 2016. The specificity of long noncoding RNA expression. Biochim. Biophys. Acta BBA – Gene Regul. Mech. 1859 (1), 16–22. https://doi.org/10.1016/j.bbagrm.2015.08.005.
Goymer, P., 2008. There's nothing abnormal about chimeric RNA 734-734 Nat. Rev. Genet. 9 (10). https://doi.org/10.1038/nrg2459.
Graça, D.S., Campagnolo, M.L., Buescu, J., 2005. Robust Simulations of Turing Machines with Analytic Maps and Flows. In: Cooper SB, Löwe B, Torenvliet L, editors. New Computational Paradigms. Berlin, Heidelberg: Springer; 2005. pp. 169–179. doi:10.1007/11494645_21.
Grass, R.N., Heckel, R., Puddu, M., Paunescu, D., Stark, W.J., 2015. Robust chemical preservation of digital information on DNA in silica with error-correcting codes. Angew. Chem. Int. Ed. 54 (8), 2552–2555. https://doi.org/10.1002/anie.201411378.
Graur, D., Zheng, Y., Price, N., Azevedo, R.B.R., Zufall, R.A., Elhaik, E., 2013. On the immortality of television sets: "function" in the human genome according to the evolution-free gospel of ENCODE. Genome Biol Evol. 5 (3), 578–590. https://doi.org/10.1093/gbe/evt028.
Graur, D., 2017. An Upper Limit on the Functional Fraction of the Human Genome. Genome Biol Evol. 2017;9: 1880–1885. doi:10.1093/gbe/evx121.
Graves, A., Wayne, G., Danihelka, I., 2014. Neural Turing Machines. ArXiv14105401 Cs. 2014 [cited 24 Sep 2020]. Available: http://arxiv.org/abs/1410.5401
Gruber, A.R., Lorenz, R., Bernhart, S.H., Neuböck, R., Hofacker, I.L., 2008. The Vienna RNA Websuite. Nucleic Acids Res. 36, W70–W74. https://doi.org/10.1093/nar/gkn188.
Guet, C.C., Bruneaux L, Min TL, Siegal-Gaskins D, Figueroa I, Emonet T, et al., 2008. Minimally invasive determination of mRNA concentration in single living bacteria. Nucleic Acids Res. 2008;36: e73. doi:10.1093/nar/gkn329.
Guttman, M., Amit, I., Garber, M., French, C., Lin, M.F., Feldser, D., Huarte, M., Zuk, O. r., Carey, B.W., Cassady, J.P., Cabili, M.N., Jaenisch, R., Mikkelsen, T.S., Jacks, T., Hacohen, N., Bernstein, B.E., Kellis, M., Regev, A., Rinn, J.L., Lander, E.S., 2009. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. Nature 458 (7235), 223–227. https://doi.org/10.1038/nature07672.
Guvenek, A., Tian, B., 2018. Analysis of alternative cleavage and polyadenylation in mature and differentiating neurons using RNA-seq data. Quant Biol. 6 (3), 253–266. https://doi.org/10.1007/s40484-018-0148-3.
Hamilton, M.L., 2012. (Monica L. Conservation of exon scrambling in human and mouse. Thesis, Massachusetts Institute of Technology. 2012. Available: https://dspace.mit.edu/handle/1721.1/72822.
Hechter, O., Halkerston, I.D.K., 1964. On the nature of macromolecular coding in neuronal memory. Perspect. Biol. Med. 7 (2), 183–198. https://doi.org/10.1353/pbm.1964.0030.
Hedrich, R., Neher, E., 2018. Venus flytrap: how an excitable, carnivorous plant works. Trends Plant Sci. 23 (3), 220–234. https://doi.org/10.1016/j.tplants.2017.12.004.
Higgs, P.G., Lehman, N., 2015. The RNA World: molecular cooperation at the origins of life. Nat. Rev. Genet. 16 (1), 7–17. https://doi.org/10.1038/nrg3841.
Hjelmfelt, A., Weinberger, E.D., Ross, J., 1991. Chemical implementation of neural networks and Turing machines. Proc. Natl. Acad. Sci. 88 (24), 10983–10987. https://doi.org/10.1073/pnas.88.24.10983.
Hnilicová, J., Staněk, D., 2011. Where splicing joins chromatin. Nucleus. 2 (3), 182–188. https://doi.org/10.4161/nucl.2.3.15876.
Hon, C.-C., Ramilowski, J.A., Harshbarger, J., Bertin, N., Rackham, O.J.L., Gough, J., Denisenko, E., Schmeier, S., Poulsen, T.M., Severin, J., Lizio, M., Kawaji, H., Kasukawa, T., Itoh, M., Burroughs, A.M., Noma, S., Djebali, S., Alam, T., Medvedeva, Y.A., Testa, A.C., Lipovich, L., Yip, C.-W., Abugessaisa, I., Mendez, M., Hasegawa, A., Tang, D., Lassmann, T., Heutink, P., Babina, M., Wells, C.A., Kojima, S., Nakamura, Y., Suzuki, H., Daub, C.O., de Hoon, M.J.L., Arner, E., Hayashizaki, Y., Carninci, P., Forrest, A.R.R., 2017. An atlas of human long non-coding RNAs with accurate 5′ ends. Nature 543 (7644), 199–204. https://doi.org/10.1038/nature21374.
Horiuchi, T., Aigaki, T., 2006. Alternative trans-splicing: a novel mode of pre-mRNA processing. Biol. Cell 98, 135–140. https://doi.org/10.1042/BC20050002.
Huranová, M., Ivani, I., Benda, A., Poser, I., Brody, Y., Hof, M., Shav-Tal, Y., Neugebauer, K.M., Staněk, D., 2010. The differential interaction of snRNPs with pre-mRNA reveals splicing kinetics in living cells. J. Cell Biol. 191 (1), 75–86. https://doi.org/10.1083/jcb.20100403010.1083/jcb.201004030.dv.
Hydén, H., 1961. Satellite cells in the nervous system. Sci. Am. 205 (6), 62–70. https://doi.org/10.1038/scientificamerican1261-62.
Jain, N., Blauch, L.R., Szymanski, M.R., Das, R., Tang, S.K.Y., Yin, Y.W., Fire, A.Z., 2020. Transcription polymerase–catalyzed emergence of novel RNA replicons. Science 368 (6487).
Jirenhed, D.-A., Rasmussen, A., Johansson, F., Hesslow, G., 2017. Learned response sequences in cerebellar Purkinje cells. Proc Natl Acad Sci U S A. 114 (23), 6127–6132. https://doi.org/10.1073/pnas.1621132114.
Johansson, F., Jirenhed, D.-A., Rasmussen, A., Zucca, R., Hesslow, G., 2014. Memory trace and timing mechanism localized to cerebellar Purkinje cells. Proc Natl Acad Sci U S A. 111 (41), 14930–14934. https://doi.org/10.1073/pnas.1415371111.
Johansson, F., Carlsson, H.E., Rasmussen, A., Yeo, C., Hesslow, G., 2015. Activation of a Temporal Memory in Purkinje Cells by the mGluR7 Receptor. Cell Rep. 13 (9), 1741–1746. https://doi.org/10.1016/j.celrep.2015.10.047.
Kakimoto, Y., Fujinuma, A., Fujita, S., Kikuchi, Y., Umekage, S., 2015. Abnormal rapid non-linear RNA production induced by T7 RNA polymerase in the absence of an exogenous DNA template. AIP Conf. Proc. 1649, 113–115. https://doi.org/10.1063/1.4913554.
Kapranov, P., St Laurent, G., Raz, T., Ozsolak, F., Reynolds, C.P., Sorensen, P.HB., Reaman, G., Milos, P., Arceci, R.J., Thompson, J.F., Triche, T.J., 2010. The majority of total nuclear-encoded non-ribosomal RNA in a human cell is "dark matter" un-annotated RNA. BMC Biol. 8 (1). https://doi.org/10.1186/1741-7007-8-149.
Kapranov, P., St. Laurent, G., 2012. Dark matter RNA: existence, function, and controversy. Front. Genet. 3. https://doi.org/10.3389/fgene.2012.00060.
Kapranov, P., Ozsolak, F., Kim, S.W., Foissac, S., Lipson, D., Hart, C., Roels, S., Borel, C., Antonarakis, S.E., Monaghan, A.P., John, B., Milos, P.M., 2010. New class of gene-termini-associated human RNAs suggests a novel RNA copying mechanism. Nature 466 (7306), 642–646. https://doi.org/10.1038/nature09190.
Khalil, A.M., Guttman, M., Huarte, M., Garber, M., Raj, A., Rivea Morales, D., Thomas, K., Presser, A., Bernstein, B.E., van Oudenaarden, A., Regev, A., Lander, E.S., Rinn, J.L., 2009. Many human large intergenic noncoding RNAs associate with chromatin-modifying complexes and affect gene expression. Proc. Natl. Acad. Sci. 106 (28), 11667–11672. https://doi.org/10.1073/pnas.0904715106.
Kim, J., Zhao, B., Huang, A.Y., Miller, M.B., Lodato, M.A., Walsh, C.A., Lee, E.A., 2020. APP gene copy number changes reflect exogenous contamination. Nature 584 (7821), E20–E28. https://doi.org/10.1038/s41586-020-2522-3.
Kleene, S.C., 1936. λ-definability and recursiveness. Duke Math J. 2, 340–353. https://doi.org/10.1215/S0012-7094-36-00227-2.
Kleinman, C.L., Majewski, J., 2012. Comment on "Widespread RNA and DNA sequence differences in the human transcriptome". Science 335 (6074). 1302-1302.
Knoll, A.H., 2011. The multiple origins of complex multicellularity. Annu. Rev. Earth Planet. Sci. 39 (1), 217–239. https://doi.org/10.1146/annurev.earth.031208.100209.
Koiran, P., Cosnard, M., Garzon, M., 1994. Computability with low-dimensional dynamical systems. Theor Comput Sci. 132 (1-2), 113–128. https://doi.org/10.1016/0304-3975(94)90229-1.
Koiran, P., Moore, C., 1999. Closed-form analytic maps in one and two dimensions can simulate universal turing machines. Theor Comput Sci. 210 (1), 217–223. https://doi.org/10.1016/S0304-3975(98)00117-0.
Konarska, M.M., Sharp, P.A., 1989. Replication of RNA by the DNA-dependent RNA polymerase of phage T7. Cell 57 (3), 423–431. https://doi.org/10.1016/0092-8674(89)90917-3.
Konarska, M.M., Sharp, P.A., 1990. Structure of RNAs replicated by the DNA-dependent T7 RNA polymerase. Cell 63 (3), 609–618. https://doi.org/10.1016/0092-8674(90)90456-O.

Kong, R., 2005. Studies of Exon Scrambling and Mutually Exclusive Alternative Splicing. Thesis. Massachusetts Institute of Technology. Available: https://dspace.mit.edu/handle/1721.1/28938.
Koš, M., Tollervey, D., 2010. Yeast pre-rRNA processing and modification occur cotranscriptionally. Mol. Cell 37 (6), 809–820. https://doi.org/10.1016/j.molcel.2010.02.024.
Kutter, C., Watt, S., Stefflova, K., Wilson, M.D., Goncalves, A., Ponting, C.P., Odom, D. T., Marques, A.C., Bartel, D.P., 2012. Rapid turnover of long noncoding RNAs and the evolution of gene expression. PLoS Genet. 8 (7), e1002841. https://doi.org/10.1371/journal.pgen.1002841.
Lafitte, G., Weiss, M., 2008. Computability of Tilings. In: Ausiello, G., Karhumäki, J., Mauri, G., Ong, L. (Eds.), IFIP International Federation for Information ProcessingFifth Ifip International Conference On Theoretical Computer Science – Tcs 2008. Springer US, Boston, MA, pp. 187–201.
Lai, M.M.C., 2005. RNA replication without RNA-dependent RNA polymerase: surprises from hepatitis delta virus. J. Virol. 79 (13), 7951–7958. https://doi.org/10.1128/JVI.79.13.7951-7958.2005.
Lakin, M.R., Modelling, P.A., 2011. Simulating and verifying turing-powerful strand displacement systems. In: Cardelli, L., Shih, W. (Eds.), DNA Computing and Molecular Programming. Springer, Berlin, Heidelberg, pp. 130–144. https://doi.org/10.1007/978-3-642-23638-9_12.
Landauer, T.K., 1964. Two hypotheses concerning the biochemical basis of memory. Psychol. Rev. 71 (3), 167–179. https://doi.org/10.1037/h0044170.
Landenmark, H.K.E., Forgan, D.H., Cockell, C.S., 2015. An estimate of the Total DNA in the biosphere. PLoS Biol. 13 (6), e1002168. https://doi.org/10.1371/journal.pbio.1002168.
Langille, J.J., Gallistel, C.R., 2020. Locating the engram: should we look for plastic synapses or information-storing molecules? Neurobiol. Learn. Mem. 169, 107164. https://doi.org/10.1016/j.nlm.2020.107164.
Lee M-H, Chun J. Mosaic APP Gene Recombination in Alzheimer's Disease—What's Next? J Exp Neurosci. 2019;13. doi:10.1177/1179069519849669.
Lee, M.-H., Liu, C.S., Zhu, Y., Kaeser, G.E., Rivera, R., Romanow, W.J., Kihara, Y., Chun, J., 2020. Reply to: APP gene copy number changes reflect exogenous contamination. Nature 584 (7821), E29–E33. https://doi.org/10.1038/s41586-020-2523-2.
Lee, J.Y., Shin, S.-Y., Park, T.H., Zhang, B.-T., 2004. Solving traveling salesman problems with DNA molecules encoding numerical values. Biosystems. 78 (1-3), 39–47. https://doi.org/10.1016/j.biosystems.2004.06.005.
Lee, M.-H., Siddoway, B., Kaeser, G.E., Segota, I., Rivera, R., Romanow, W.J., Liu, C.S., Park, C., Kennedy, G., Long, T., Chun, J., 2018. Somatic APP gene recombination in Alzheimer's disease and normal neurons. Nature 563 (7733), 639–645. https://doi.org/10.1038/s41586-018-0718-6.
Lee, H., Zhang, Z., Krause, H.M., 2019. Long noncoding RNAs and repetitive elements: junk or intimate evolutionary partners? Trends Genet. 35 (12), 892–902. https://doi.org/10.1016/j.tig.2019.09.006.
Lehmann, E., Brueckner, F., Cramer, P., 2007. Molecular basis of RNA-dependent RNA polymerase II activity. Nature 450 (7168), 445–449. https://doi.org/10.1038/nature06290.
Levin, Michael, 2014. Endogenous bioelectrical networks store non-genetic patterning information during development and regeneration. The Journal of Physiology 592 (11), 2295–2305. https://doi.org/10.1113/jphysiol.2014.271940.
Li, R., Ren, X., Ding, Q., Bi, Y.u., Xie, D., Zhao, Z., 2020. Direct full-length RNA sequencing reveals unexpected transcriptome complexity during Caenorhabditis elegans development. Genome Res. 30 (2), 287–298. https://doi.org/10.1101/gr.251512.119.
Li, H., Wang, J., Mor, G., Sklar, J., 2008. A neoplastic gene fusion mimics trans-splicing of RNAs in normal human cells. Science 321 (5894), 1357–1361.
Li, M., Wang, I.X., Li, Y., Bruzel, A., Richards, A.L., Toung, J.M., Cheung, V.G., 2011. Widespread RNA and DNA sequence differences in the human transcriptome. Science 333 (6038), 53–58.
Li, M., Wang, I.X., Cheung, V.G., 2012. Response to comments on "Widespread RNA and DNA sequence differences in the human transcriptome". Science 335 (6074). 1302-1302.
Liddicoat, B.J., Piskol, R., Chalk, A.M., Ramaswami, G., Higuchi, M., Hartner, J.C., Li, J. B., Seeburg, P.H., Walkley, C.R., 2015. RNA editing by ADAR1 prevents MDA5 sensing of endogenous dsRNA as nonself. Science 349 (6252), 1115–1120.
Lim, D., Maas, W.K., 1989. Reverse transcriptase in bacteria. Mol. Microbiol. 3 (8), 1141–1144. https://doi.org/10.1111/j.1365-2958.1989.tb00264.x.
Lin, W., Piskol, R., Tan, M.H., Li, J.B., 2012. Comment on "Widespread RNA and DNA sequence differences in the human transcriptome". Science 335 (6074). 1302-1302.
Linquist, S., Doolittle, W.F., Palazzo, A.F., Flint, J., 2020. Getting clear about the F-word in genomics. PLoS Genet. 16 (4), e1008702. https://doi.org/10.1371/journal.pgen.1008702.
Lipton, R.J., 1995. DNA solution of hard computational problems. Science 268 (5210), 542–545.
Liu, G., Mattick, J., Taft, R.J., 2013. A meta-analysis of the genomic and transcriptomic composition of complex life. Cell Cycle 12 (13), 2061–2072. https://doi.org/10.4161/cc.25134.
Liu, Q., Wang, L., Frutos, A.G., Condon, A.E., Corn, R.M., Smith, L.M., 2000. DNA computing on surfaces. Nature 403 (6766), 175–179. https://doi.org/10.1038/35003155.
Lorenz, R., Bernhart, S.H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P.F., Hofacker, I.L., 2011. ViennaRNA Package 2.0. Algorithms Mol. Biol. 6 (1). https://doi.org/10.1186/1748-7188-6-26.
Lu, Z., Zhang, Q., Lee, B., Flynn, R., Smith, M., Robinson, J., Davidovich, C., Gooding, A., Goodrich, K., Mattick, J., Mesirov, J., Cech, T., Chang, H., 2016. RNA duplex map in living cells reveals higher-order transcriptome structure. Cell 165 (5), 1267–1279. https://doi.org/10.1016/j.cell.2016.04.028.
Macia, A., Widmann, T.J., Heras, S.R., Ayllon, V., Sanchez, L., Benkaddour-Boumzaouad, M., Muñoz-Lopez, M., Rubio, A., Amador-Cubero, S., Blanco-Jimenez, E., Garcia-Castro, J., Menendez, P., Ng, P., Muotri, A.R., Goodier, J.L., Garcia-Perez, J.L., 2017. Engineered LINE-1 retrotransposition in nondividing human neurons. Genome Res. 27 (3), 335–348. https://doi.org/10.1101/gr.206805.116.
Magnasco, M.O., 1997. Chemical kinetics is turing universal. Phys. Rev. Lett. 78 (6), 1190–1193. https://doi.org/10.1103/PhysRevLett.78.1190.
Malinen, A.M., Turtola, M., Parthiban, M., Vainonen, L., Johnson, M.S., Belogurov, G. A., 2012. Active site opening and closure control translocation of multisubunit RNA polymerase. Nucleic Acids Res. 40, 7442–7451. https://doi.org/10.1093/nar/gks383.
Mao, C., LaBean, T.H., Reif, J.H., Seeman, N.C., 2000. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. Nature 407 (6803), 493–496. https://doi.org/10.1038/35035038.
Marek, M.S., Johnson-Buck, A., Walter, N.G., 2011. The shape-shifting quasispecies of RNA: One sequence, many functional folds. PCCP 13, 11524–11537. https://doi.org/10.1039/c1cp20576e.
Martí, E., Pantano, L., Bañez-Coronel, M., Llorens, F., Miñones-Moyano E, Porta S, et al., 2010. A myriad of miRNA variants in control and Huntington's disease brain regions detected by massively parallel sequencing. Nucleic Acids Res. 2010;38: 7219–7235. doi:10.1093/nar/gkq575.
Mathews, D.H., Disney, M.D., Childs, J.L., Schroeder, S.J., Zuker, M., Turner, D.H., 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proc. Natl. Acad. Sci. U.S.A. 101 (19), 7287–7292. https://doi.org/10.1073/pnas.0401799101.
Mattick, J.S., 2004. RNA regulation: a new genetics? Nat. Rev. Genet. 5 (4), 316–323. https://doi.org/10.1038/nrg1321.
Mattick, J.S., 2009. Has evolution learnt how to learn? 665-665 EMBO Rep 10 (7). https://doi.org/10.1038/embor.2009.135.
Mattick, J.S., 2018. The state of long non-coding RNA biology. Non-Coding RNA 4, 17. https://doi.org/10.3390/ncrna4030017.
Mattick, J.S., Dinger, M.E., 2013. The extent of functionality in the human genome. HUGO J. 7, 2. https://doi.org/10.1186/1877-6566-7-2.
Mattick, J.S., Mehler, M.F., 2008. RNA editing, DNA recoding and the evolution of human cognition. Trends Neurosci. 31 (5), 227–233. https://doi.org/10.1016/j.tins.2008.02.003.
McConnell, M.J., Lindberg, M.R., Brennand, K.J., Piper, J.C., Voet, T., Cowing-Zitron, C., Shumilina, S., Lasken, R.S., Vermeesch, J.R., Hall, I.M., Gage, F.H., 2013. Mosaic copy number variation in human neurons. Science 342 (6158), 632–637.
McFleder, R.L., Mansur, F., Richter, J.D., 2017. Dynamic control of dendritic mRNA expression by CNOT7 regulates synaptic efficacy and higher cognitive function. Cell Rep. 20 (3), 683–696. https://doi.org/10.1016/j.celrep.2017.06.078.
McLaughlin, Kelly A., Levin, Michael, 2018. Bioelectric signaling in regeneration: Mechanisms of ionic controls of growth and form. Developmental Biology 433 (2), 177–189. https://doi.org/10.1016/j.ydbio.2017.08.032.
Mercer, T.R., Dinger, M.E., Sunkin, S.M., Mehler, M.F., Mattick, J.S., 2008. Specific expression of long noncoding RNAs in the mouse brain. Proc. Natl. Acad. Sci. U.S. A. 105 (2), 716–721. https://doi.org/10.1073/pnas.0706729105.
Milo, R., Jorgensen, P., Moran, U., Weber, G., Springer, M., 2010. BioNumbers—the database of key numbers in molecular and cell biology. Nucleic Acids Res. 38, D750–D753. https://doi.org/10.1093/nar/gkp889.
Miura, P., Shenker, S., Andreu-Agullo, C., Westholm, J.O., Lai, E.C., 2013. Widespread and extensive lengthening of 3′ UTRs in the mammalian brain. Genome Res. 23 (5), 812–825. https://doi.org/10.1101/gr.146886.112.
Mogensen, T.Æ., 2001. An investigation of compact and efficient number representations in the pure lambda calculus. In: Bjørner, D., Broy, M., Zamulin, A.V. (Eds.), Perspectives of System Informatics. Springer, Berlin, Heidelberg, pp. 205–213. https://doi.org/10.1007/3-540-45575-2_20.
Mohanraju, P., Makarova, K.S., Zetsche, B., Zhang, F., Koonin, E.V., van der Oost, J., 2016. Diverse evolutionary roots and mechanistic variations of the CRISPR-Cas systems. Science 353 (6299).
Moore, C., 1990. Unpredictability and undecidability in dynamical systems. Phys. Rev. Lett. 64 (20), 2354–2357. https://doi.org/10.1103/PhysRevLett.64.2354.
Moore, C., 1991. Generalized shifts: unpredictability and undecidability in dynamical systems. Nonlinearity 4 (2), 199–230. https://doi.org/10.1088/0951-7715/4/2/002.
Moore, C., 1998. Finite-dimensional analog computers: flows, maps, and recurrent neural networks. In: 1st International Conference on Unconventional Models of Computation-UMC, pp. 59–71.
Moran, Y., Barzilai, M.G., Liebeskind, B.J., Zakon, H.H., Anderson, P.A.V., 2015. Evolution of voltage-gated ion channels at the emergence of Metazoa. J. Exp. Biol. 218 (4), 515–525. https://doi.org/10.1242/jeb.110270.
Moroz, L.L., 2009. On the independent origins of complex brains and neurons. Brain Behav. Evol. 74 (3), 177–190. https://doi.org/10.1159/000258665.
Moroz, L.L., Kohn, A.B., 2016. Independent origins of neurons and synapses: insights from ctenophores. Philos Trans R Soc B Biol Sci. 371 (1685), 20150041. https://doi.org/10.1098/rstb.2015.0041.
Mustoe, A.M., Busan, S., Rice, G.M., Hajdin, C.E., Peterson, B.K., Ruda, V.M., Kubica, N., Nutiu, R., Baryza, J.L., Weeks, K.M., 2018. Pervasive regulatory functions of

mRNA structure revealed by high-resolution SHAPE probing. Cell 173 (1), 181–195.e18. https://doi.org/10.1016/j.cell.2018.02.034.
Nigro, J.M., Cho, K.R., Fearon, E.R., Kern, S.E., Ruppert, J.M., Oliner, J.D., Kinzler, K.W., Vogelstein, B., 1991. Scrambled exons. Cell 64 (3), 607–613. https://doi.org/10.1016/0092-8674(91)90244-S.
Niklas, K.J., Newman, S.A., 2013. The origins of multicellular organisms. Evol Dev. 15 (1), 41–52. https://doi.org/10.1111/ede.12013.
Niu, D.-K., Jiang, L.i., 2013. Can ENCODE tell us how much junk DNA we carry in our genome? Biochem. Biophys. Res. Commun. 430 (4), 1340–1343. https://doi.org/10.1016/j.bbrc.2012.12.074.
Nowacki, M., Vijayan, V., Zhou, Y.i., Schotanus, K., Doak, T.G., Landweber, L.F., 2008. RNA-mediated epigenetic programming of a genome-rearrangement pathway. Nature 451 (7175), 153–158. https://doi.org/10.1038/nature06452.
Oku, M., Aihara, K., 2010. Noise-robust realization of Turing-complete cellular automata by using neural networks with pattern representation. Phys. Lett. A 374 (48), 4859–4863. https://doi.org/10.1016/j.physleta.2010.10.024.
Olofsson, S.-O., Boström, K., Carlsson, P., Borén, J., Wettesten, M., Bjursell, G., Wiklund, O., Bondjers, G., 1987. Structure and biosynthesis of apolipoprotein B. Am. Heart J. 113 (2), 446–452. https://doi.org/10.1016/0002-8703(87)90612-0.
Owens, M.C., Clark, S.C., Yankey, A., Somarowthu, S., 2019. Identifying structural domains and conserved regions in the long non-coding RNA lncTCF7. Int. J. Mol. Sci. 20 (19), 4770. https://doi.org/10.3390/ijms20194770.
Palazzo, A.F., Gregory, T.R., Akey, J.M., 2014. The case for junk DNA. PLoS Genet. 10 (5), e1004351. https://doi.org/10.1371/journal.pgen.1004351.
Palazzo, A.F., Lee, E.S., 2015. Non-coding RNA: what is functional and what is junk? Front. Genet. 6. https://doi.org/10.3389/fgene.2015.00002.
Papadimitriou, C.H., Vempala, S.S., Mitropolsky, D., Collins, M., Maass, W., 2020. Brain computation by assemblies of neurons. Proc. Natl. Acad. Sci. 117, 14464–14472. https://doi.org/10.1073/pnas.2001893117.
Pastuzyn, E.D., Day, C.E., Kearns, R.B., Kyrke-Smith, M., Taibi, A.V., McCormick, J., Yoder, N., Belnap, D.M., Erlendsson, S., Morado, D.R., Briggs, J.A.G., Feschotte, C., Shepherd, J.D., 2018. The neuronal gene arc encodes a repurposed retrotransposon gag protein that mediates intercellular RNA transfer. Cell 172 (1-2), 275–288.e18. https://doi.org/10.1016/j.cell.2017.12.024.
Pauli, A., Valen, E., Lin, M.F., Garber, M., Vastenhouw, N.L., Levin, J.Z., Fan, L., Sandelin, A., Rinn, J.L., Regev, A., Schier, A.F., 2012. Systematic identification of long noncoding RNAs expressed during zebrafish embryogenesis. Genome Res. 22 (3), 577–591. https://doi.org/10.1101/gr.133009.111.
Pearce K, Cai D, Roberts AC, Glanzman DL. Role of protein synthesis and DNA methylation in the consolidation and maintenance of long-term memory in Aplysia. Ramaswami M, editor. eLife. 2017;6: e18299. doi:10.7554/eLife.18299.
Pérez, J., Marinković, J., Barceló, P., 2019. On the Turing Completeness of Modern Neural Network Architectures. ArXiv190103429 Cs Stat. 2019 [cited 28 Feb 2021]. Available: http://arxiv.org/abs/1901.03429.
Pheasant, M., Mattick, J.S., 2007. Raising the estimate of functional human sequences. Genome Res. 17, 1245–1253. https://doi.org/10.1101/gr.6406307.
Piantadosi, S.T., 2021. The computational origin of representation. Minds Mach. 31 (1), 1–58. https://doi.org/10.1007/s11023-020-09540-9.
Pickrell, J.K., Gilad, Y., Pritchard, J.K., 2012. Comment on "Widespread RNA and DNA sequence differences in the human transcriptome". Science 335 (6074). 1302-1302.
Piovesan, A., Caracausi, M., Antonaros, F., Pelleri, M.C., Vitale, L., 2016. GeneBase 1.1: a tool to summarize data from NCBI gene datasets and its application to an update of human gene statistics. Database J. Biol. Databases Curation. 2016; doi:10.1093/database/baw153.
Ponjavic, J., Ponting, C.P., Lunter, G., 2007. Functionality or transcriptional noise? evidence for selection within long noncoding RNAs. Genome Res. 17 (5), 556–565. https://doi.org/10.1101/gr.6036807.
Ponting, C.P., Hardison, R.C., 2011. What fraction of the human genome is functional? Genome Res. 21 (11), 1769–1776. https://doi.org/10.1101/gr.116814.110.
Poo, M.-M., Pignatelli, M., Ryan, T.J., Tonegawa, S., Bonhoeffer, T., Martin, K.C., Rudenko, A., Tsai, L.-H., Tsien, R.W., Fishell, G., Mullins, C., Gonçalves, J.T., Shtrahman, M., Johnston, S.T., Gage, F.H., Dan, Y., Long, J., Buzsáki, G., Stevens, C., 2016. What is memory? the present state of the engram. BMC Biol. 14 (1). https://doi.org/10.1186/s12915-016-0261-6.
Qian, L., Soloveichik, D., Winfree, E., 2011. Efficient turing-universal computation with DNA polymers. In: Sakakibara, Y., Mi, Y. (Eds.), DNA Computing and Molecular Programming. Springer, Berlin, Heidelberg, pp. 123–140. https://doi.org/10.1007/978-3-642-18305-8_12.
Qian, L., Winfree, E., Bruck, J., 2011. Neural network computation with DNA strand displacement cascades. Nature 475 (7356), 368–372. https://doi.org/10.1038/nature10262.
Queenan, B.N., Ryan, T.J., Gazzaniga, M., Gallistel, C.R., 2017. On the research of time past: the hunt for the substrate of memory. Ann. N. Y. Acad. Sci. 1396, 108–125. https://doi.org/10.1111/nyas.13348.
Rands, C.M., Meader, S., Ponting, C.P., Lunter, G., Schierup, M.H., 2014. 8.2% of the human genome is constrained: variation in rates of turnover across functional element classes in the human lineage. PLoS Genet. 10 (7), e1004525. https://doi.org/10.1371/journal.pgen.1004525.
Ravasi, T., Suzuki, H., Pang, K.C., Katayama, S., Furuno, M., Okunishi, R., Fukuda, S., Ru, K., Frith, M.C., Gongora, M.M., Grimmond, S.M., Hume, D.A., Hayashizaki, Y., Mattick, J.S., 2006. Experimental validation of the regulated expression of large numbers of non-coding RNAs from the mouse genome. Genome Res. 16 (1), 11–19. https://doi.org/10.1101/gr.4200206.
Reif, J.H., Tygar, J.D., Yoshida, A., 1990. The computability and complexity of optical beam tracing. Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science. 1990. pp. 106–114 vol.1. doi:10.1109/FSCS.1990.89529.
Rendell, P., 2002. Turing Universality of the Game of Life. In: Adamatzky, A. (Ed.), Collision-Based Computing. Springer, London, pp. 513–539. https://doi.org/10.1007/978-1-4471-0129-1_18.
Rigatti, R., Jia, J.-H., Samani, N.J., Eperon, I.C., 2004. Exon repetition: a major pathway for processing mRNA of some genes is allele-specific. Nucleic Acids Res. 32, 441–446. https://doi.org/10.1093/nar/gkh197.
Rohrback, S., Siddoway, B., Liu, C.S., Chun, J., 2018. Genomic mosaicism in the developing and adult brain. Dev Neurobiol. 78 (11), 1026–1048. https://doi.org/10.1002/dneu.22626.
Rothemund, P.W.K., 1996. A DNA and restriction enzyme implementation of Turing Machines. In: Lipton, R.J., Baum, E.B. (Eds.), DNA based computers. American Mathematical Society, Providence, RI, pp. 75–120. Available: https://resolver.caltech.edu/CaltechAUTHORS:20111024-134806267.
Rothemund, P.W.K., Papadakis, N., Winfree, E., Anne Condon, 2004. Algorithmic self-assembly of DNA sierpinski triangles. PLoS Biol. 2 (12), e424. https://doi.org/10.1371/journal.pbio.0020424.
Ruben, A.J., Landweber, L.F., 2000. The past, present and future of molecular computing. Nat. Rev. Mol. Cell Biol. 1 (1), 69–72. https://doi.org/10.1038/35036086.
Rumelhart, David E., McClelland, James L.PDP Research Group, 1986. Parallel distributed processing: explorations in the microstructure of cognition, vol. 2: psychological and biological models, 2. MIT Press.
Ryan, T.J., Roy, D.S., Pignatelli, M., Arons, A., Tonegawa, S., 2015. Engram cells retain memory under retrograde amnesia. Science 348 (6238), 1007–1013.
Sakamoto, K., Gouzu, H., Komiya, K., Kiga, D., Yokoyama, S., Yokomori, T., Hagiya, M., 2000. Molecular computation by DNA hairpin formation. Science 288 (5469), 1223–1226.
Samson, J., Cronin, S., Dean, K., 2018. BC200 (BCYRN1) – the shortest, long, non-coding RNA associated with cancer. Non-Coding RNA Res. 3 (3), 131–143. https://doi.org/10.1016/j.ncrna.2018.05.003.
Santin, J.M., Schulz, D.J., 2019. Membrane voltage is a direct feedback signal that influences correlated ion channel expression in neurons. Curr. Biol. 29 (10), 1683–1688.e2. https://doi.org/10.1016/j.cub.2019.04.008.
Scarle, S., 2009. Implications of the Turing completeness of reaction-diffusion models, informed by GPGPU simulations on an XBox 360: Cardiac arrhythmias, re-entry and the Halting problem. Comput. Biol. Chem. 33 (4), 253–260. https://doi.org/10.1016/j.compbiolchem.2009.05.001.
Schmitt, F.O., 1966. Molecular and ultrastructural correlates of function in neurons, neuronal nets, and the brain. Naturwissenschaften 53 (3), 71–79. https://doi.org/10.1007/BF00594750.
Schmitt, F.O., 1967. Molecular biology among the neurosciences. Arch. Neurol. 17 (6), 561–572. https://doi.org/10.1001/archneur.1967.00470300003001.
Schmitt, F.O., 1962. Macromolecular Specificity and Biological Memory | The MIT Press. 1962 [cited 27 Apr 2020]. Available: https://mitpress.mit.edu/books/macromolecular-specificity-and-biological-memory.
Schönfinkel, M., 1924. Über die Bausteine der mathematischen Logik. Math Ann. 1924;92: 305–316. doi:10.1007/BF01448013.
Shao, X., Shepelev, V., Fedorov, A., 2006. Bioinformatic analysis of exon repetition, exon scrambling and trans-splicing in humans. Bioinforma Oxf. Engl. 22 (6), 692–698. https://doi.org/10.1093/bioinformatics/bti795.
Shapiro, E., 2012. A mechanical Turing machine: blueprint for a biomolecular computer. Interface Focus. 2 (4), 497–503. https://doi.org/10.1098/rsfs.2011.0118.
Shapiro, E., Karunaratne, K.S.G., 2001. Method and system of computing similar to a turing machine. US6266569B1, 2001. Available: https://patents.google.com/patent/US6266569B1/en.
Shapiro, E., Benenson, Y., 2006. Bringing DNA computers to life. Sci. Am. 294 (5), 44–51.
Shapiro, E., Gil, B., 2008. RNA computing in a living cell. Science 322 (5900), 387–388.
Siegelmann, H.T., Sontag, E.D., 1991. Turing computability with neural nets. Appl. Math. Lett. 4 (6), 77–80. https://doi.org/10.1016/0893-9659(91)90080-F.
Šíma, J., Orponen, P., 2003. General-purpose computation with neural networks: a survey of complexity theoretic results. Neural Comput. 15 (12), 2727–2778.
Simon, D.M., Zimmerly, S., 2008. A diversity of uncharacterized reverse transcriptases in bacteria. Nucleic Acids Res. 36, 7219–7229. https://doi.org/10.1093/nar/gkn867.
Singh S, Qin F, Kumar S, Elfman J, Lin E, Pham L-P, et al., 2020. The landscape of chimeric RNAs in non-diseased tissues and cells. Nucleic Acids Res. 2020;48: 1764–1778. doi:10.1093/nar/gkz1223.
Siuti, P., Yazbek, J., Lu, T.K., 2013. Synthetic circuits integrating logic and memory in living cells. Nat. Biotechnol. 31 (5), 448–452. https://doi.org/10.1038/nbt.2510.
Siwiak, M., Zielenkiewicz, P., Levy, Y.K., 2013. Transimulation – protein biosynthesis web service. PLoS ONE 8 (9), e73943. https://doi.org/10.1371/journal.pone.0073943.
Smith, M.A., Gesell, T., Stadler, P.F., Mattick, J.S., 2013. Widespread purifying selection on RNA structure in mammals. Nucleic Acids Res. 41, 8220–8236. https://doi.org/10.1093/nar/gkt596.

Soloveichik, D., Cook, M., Winfree, E., Bruck, J., 2008. Computation with finite stochastic chemical reaction networks. Nat. Comput. 7 (4), 615–633. https://doi.org/10.1007/s11047-008-9067-y.
Soloveichik, D., Seelig, G., Winfree, E., 2010. DNA as a universal substrate for chemical kinetics. Proc. Natl. Acad. Sci. 107 (12), 5393–5398. https://doi.org/10.1073/pnas.0909380107.
Somarowthu, S., Legiewicz, M., Chillón, I., Marcia, M., Liu, F., Pyle, A., 2015. HOTAIR forms an intricate and modular secondary structure. Mol. Cell 58 (2), 353–361. https://doi.org/10.1016/j.molcel.2015.03.006.
Sone, M., Hayashi, T., Tarui, H., Agata, K., Takeichi, M., Nakagawa, S., 2007. The mRNA-like noncoding RNA Gomafu constitutes a novel nuclear domain in a subset of neurons. J. Cell Sci. 120, 2498–2506. https://doi.org/10.1242/jcs.009357.
St. Laurent, G., Vyatkin, Y., Kapranov, P., 2014. Dark matter RNA illuminates the puzzle of genome-wide association studies. BMC Med. 12 (1). https://doi.org/10.1186/1741-7015-12-97.
Stephen, S., Pheasant, M., Makunin, I.V., Mattick, J.S., 2008. Large-scale appearance of ultraconserved elements in tetrapod genomes and slowdown of the molecular clock. Mol. Biol. Evol. 25, 402–408. https://doi.org/10.1093/molbev/msm268.
Sterling, P., Laughlin, S., 2015. Principles of Neural Design. The MIT Press. 10.2307/j.ctt17kk982.
Taft, R.J., Pheasant, M., Mattick, J.S., 2007. The relationship between non-protein-coding DNA and eukaryotic complexity. BioEssays 29 (3), 288–299. https://doi.org/10.1002/bies.20544.
Tang, Y., Ma, S., Wang, X., Xing, Q.u., Huang, T., Liu, H., Li, Q., Zhang, Y., Zhang, K., Yao, M., Yang, G.L., Li, H., Zang, X., Yang, B.o., Guan, F., 2019. Identification of chimeric RNAs in human infant brains and their implications in neural differentiation. Int. J. Biochem. Cell Biol. 111, 19–26. https://doi.org/10.1016/j.biocel.2019.03.012.
Tani, H., Mizutani, R., Salam, K.A., Tano, K., Ijiri, K., Wakamatsu, A.i., Isogai, T., Suzuki, Y., Akimitsu, N., 2012. Genome-wide determination of RNA stability reveals hundreds of short-lived noncoding transcripts in mammals. Genome Res. 22 (5), 947–956. https://doi.org/10.1101/gr.130559.111.
Treiber, C.D., Waddell, S., 2017. Resolving the prevalence of somatic transposition in Drosophila. Flint J, editor. eLife. 6. https://doi.org/10.7554/eLife.28297.
Tsai, M.-C., Manor, O., Wan, Y., Mosammaparast, N., Wang, J.K., Lan, F., Shi, Y., Segal, E., Chang, H.Y., 2010. Long noncoding RNA as modular scaffold of histone modification complexes. Science 329 (5992), 689–693.
Tseng, C.-H., Lai, M.M., 2009. Hepatitis delta virus RNA replication. Viruses 1 (3), 818–831. https://doi.org/10.3390/v1030818.
Turing, A.M., 1936. On computable numbers, with an application to the Entscheidungsproblem. J. Math. 58, 5.
Turing, A.M., 1937. The þ-function in λ-K-conversion 164–164 J. Symb. Log. 2 (4). https://doi.org/10.2307/2268281.
Turing, A.M., 1937. Computability and λ-definability. J. Symb. Log. 2 (4), 153–163. https://doi.org/10.2307/2268280.
Udagawa, T., Swanger, S., Takeuchi, K., Kim, J., Nalavadi, V., Shin, J., Lorenz, L., Zukin, R.S., Bassell, G., Richter, J., 2012. Bidirectional control of mRNA translation and synaptic plasticity by the cytoplasmic polyadenylation complex. Mol. Cell 47 (2), 253–266. https://doi.org/10.1016/j.molcel.2012.05.016.
Ungar, G., 1973. The problem of molecular coding of neural information. A critical review. Naturwissenschaften. 60 (7), 307–312.
Uphouse, L.L., MacInnes, J.W., Schlesinger, K., 1974. Role of RNA and protein in memory storage: a review. Behav. Genet. 4 (1), 29–81.
Upton, K., Gerhardt, D., Jesuadian, J.S., Richardson, S., Sánchez-Luque, F., Bodea, G., Ewing, A., Salvador-Palomeque, C., van der Knaap, M., Brennan, P., Vanderver, A., Faulkner, . Ubiquitous L1 mosaicism in hippocampal neurons. Cell 161 (2), 228–239. https://doi.org/10.1016/j.cell.2015.03.026.
van Bakel, H., Nislow, C., Blencowe, B.J., Hughes, T.R., Eisen, M.B., 2011. Response to "the reality of pervasive transcription". PLoS Biol. 9 (7), e1001102. https://doi.org/10.1371/journal.pbio.1001102.
van Sickle, G.R., McCluer, R.H., 1966. Transduction of neuronal electric response to macromolecular synthesis: a theoretical construct of a fundamental unit for information storage and retrieval. Perspect. Biol. Med. 9 (3), 425–432. https://doi.org/10.1353/pbm.1966.0013.
Varghese, S., Elemans, J.A.A.W., Rowan, A.E., Nolte, R.J.M., 2015. Molecular computing: paths to chemical Turing machines. Chem. Sci. 6 (11), 6050–6058.
Wan, Y., Kertesz, M., Spitale, R.C., Segal, E., Chang, H.Y., 2011. Understanding the transcriptome through RNA structure. Nat. Rev. Genet. 12 (9), 641–655. https://doi.org/10.1038/nrg3049.
Wang, I.X., Core, L.J., Kwak, H., Brady, L., Bruzel, A., McDaniel, L., et al., 2014. RNA-DNA differences are generated in human cells within seconds after RNA exits Pol II. Cell Rep. 6, 906–915. https://doi.org/10.1016/j.celrep.2014.01.037.
Wang, F., Durfee, L., Huibregtse, J., 2013. A cotranslational ubiquitination pathway for quality control of misfolded proteins. Mol. Cell 50 (3), 368–378. https://doi.org/10.1016/j.molcel.2013.03.009.
Wang, Y., Liu, H., Sun, Z., 2017. Lamarck rises from his grave: parental environment-induced epigenetic inheritance in model organisms and humans. Biol. Rev. 92 (4), 2084–2111. https://doi.org/10.1111/brv.12322.
Washietl, S., Hofacker, I.L., Lukasser, M., Hüttenhofer, A., Stadler, P.F., 2005. Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome. Nat. Biotechnol. 23 (11), 1383–1390. https://doi.org/10.1038/nbt1144.
Westra, J.W., Rivera, R.R., Bushman, D.M., Yung, Y.C., Peterson, S.E., Barral, S., Chun, J., 2010. Neuronal DNA content variation (DCV) with regional and individual differences in the human brain. J. Comp. Neurol. 518 (19), 3981–4000. https://doi.org/10.1002/cne.22436.
Wettich, A., Biebricher, C.K., 2001. RNA species that replicate with DNA-dependent RNA polymerase from Escherichia coli. Biochemistry 40 (11), 3308–3315. https://doi.org/10.1021/bi002756g.
White, E., Schlackow, M., Kamieniarz-Gdula, K., Proudfoot, N.J., Gullerova, M., 2014. Human nuclear Dicer restricts the deleterious accumulation of endogenous double-stranded RNA. Nat. Struct. Mol. Biol. 21 (6), 552–559. https://doi.org/10.1038/nsmb.2827.
Win, M.N., Smolke, C.D., 2008. Higher-order cellular information processing with synthetic RNA devices. Science 322 (5900), 456–460.
Winfree, E., Liu, F., Wenzler, L.A., Seeman, N.C., 1998. Design and self-assembly of two-dimensional DNA crystals. Nature 394 (6693), 539–544. https://doi.org/10.1038/28998.
Woods, D., Doty, D., Myhrvold, C., Hui, J., Zhou, F., Yin, P., Winfree, E., 2019. Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. Nature 567 (7748), 366–372. https://doi.org/10.1038/s41586-019-1014-9.
Yahiro, W., Hagiya, M., 2016. Implementation of turing machine using DNA strand displacement. In: Martín-Vide, C., Mizuki, T., Vega-Rodríguez, M.A. (Eds.), Theory and Practice of Natural Computing. Springer International Publishing, Cham, pp. 161–172. https://doi.org/10.1007/978-3-319-49001-4_13.
Yoffe, A.M., Prinsen, P., Gelbart, W.M., Ben-Shaul, A., 2011. The ends of a large RNA molecule are necessarily close. Nucleic Acids Res. 39, 292–299. https://doi.org/10.1093/nar/gkq642.
Zhao, B., Sun, J., Zhang, X., Mo, H., Niu, Y., Li, Q., Wang, L., Zhong, Y.i., 2019. Long-term memory is formed immediately without the need for protein synthesis-dependent consolidation in Drosophila. Nat. Commun. 10 (1). https://doi.org/10.1038/s41467-019-12436-7.