The Natural Evolution of Computing

Salva Duran-Nebreda and Sergi Valverde

September 25, 2021

For submission to the Oxford Handbook of Cultural Evolution
Edited by R. Kendal, J. Tehrani and J. Kendal

Abstract

The evolution of computing is an example of a major, transformative technological adaptation still unfolding in human history. Information technologies are supported by many other knowledge domains that have evolved through a cumulative cultural process, yet at the same time computing affects the tempo and mode of cultural evolution, greatly accelerating innovation processes driven by recombination of present technologies. Additionally, computing has created entire new domains for cumulative cultural evolution, furthering an era dominated by digital economies and media. These new domains offer very desirable qualities for cultural evolution research and digital archaeology, including good coverage in data completeness in widely different aspects of human culture, from social networks to innovation in programming languages. We review the major transitions in information technologies, with especial interest in their connections to a biological evolutionary framework. In particular, software vs. hardware evolution poses an interesting example of symbiotic technologies that display strong social dependencies as well as an extrinsic fitness due to energetic and temporal constrains. Properly accounting for the interplay of material and social factors can explain the coexistence of gradualism and punctuated dynamics in cultural and technological evolution.

Keywords: cultural evolution, major transitions, information technology, software, hardware, computing.

1 Introduction

> In natural science, Nature has given us a world and we're just to discover its laws. In computers, we can stuff laws into it and create a world.
>
> *Alan Kay*

The evolution of computing is a major, transforming technological adaptation still unfolding in human history. Although there is no formal definition [1], our world is often described as an "information society" or "algorithmic society" [2]. Computing is a very recent historical phenomenon and its long-term implications are yet to be determined and analyzed. This is reflected in the ambivalence about the meaning of key concepts like "information technology (IT)" and "computing". The meaning of IT shifted from the "conjunction of computers, operation research methods, and simulation techniques" to encompass every information distribution technology (specially the Internet and related technologies) to be often used interchangeably with computing [3]. Still, the lack of consensus hasn't precluded computing and information technologies from having a massive impact at many levels, from social interactions to businesses and the natural environment. Some specific characteristics, like digital media, accelerate transmission and modification of cultural knowledge [4, 5], but also pose important social and environmental threats [6].

Familiarity with computing devices contrasts with our partial knowledge about the underlying evolutionary mechanisms. Laptops, tablets and smartphones are so commonplace today that they became synonymous with the word 'technology'. But as familiar as the smartphone might be, studies often relegate it as a metaphor for another natural or artificial system. This is a misleading simplification that needs to be addressed. Moreover, although there is a rich literature on the history of computing (e.g., the journal "Annals of the History of Computing"), few cultural evolution studies had focused in computing systems. One problem is that comparing computing with other technologies is quite difficult. For example, the relationship between structure and function is very different in computers and other physical artifacts, like hammers, boats or forks. In addition, studies of technology have traditionally emphasized physical artifacts, which disregards the key role of immaterial technologies in today's society [7].

We can extend cultural evolution to the study of computing and information technology. The computer that sits on your table is one of the most complex technologies developed by humans, and it is very unlikely that a single person could have created the thousands of components needed to assemble it. Instead, the existence of the computer crucially depends on the accumulation of knowledge and expertise over many years. This process of technological change can be readily conceptualized and studied from the perspective of cultural evolution (and we review here some of these ongoing efforts). But computing also involves radical innovations that are much more difficult to understand within the current framework. Hardware software separation is an unprecedented event in history, and software displays features that have been suggested to be closer to biological systems than to other engineered artifacts [8]. Indeed, programmers have long sought to mimic the features of biological systems and evolution for enhanced software reliability and evolvability [9, 10]. Is there an analog of software in older technology? How do social interactions shape computing innovations? What is the role of information technology in cultural evolution?

2 Computing in a nutshell

From a historical perspective, computing represents a continuation of a much longer trend in the evolution of cognitive tools [11]. The idea of computation is a very old one, and devices aiding or extending manual computations have been used for thousands of years (the abacus is thought to have been invented in Babylon ca 2400 BC). Moreover, the design principles of computers were known for hundred of years [12]. And yet, computer machines could not be realized (and massively adopted) until very recently. Retracing the main events in the history of computing (see Figure 1c) has the potential to reveal its co-evolution with culture. Any historical review is necessarily biased due to the exponential growth of information technology (see below) and the interested reader is pointed to the excellent literature on the history of computing, e.g., [13] and [14].

Humans have been storing, retrieving, manipulating and communicating information since the invention of writing (approx. 3000 BC [15]). Externalization of information storage together with development of mathematical notation drove down the costs of calculation, and thus providing the grounds for achieving the first analogical computers. Such devices were capable of simple arithmetic calculations by exploiting the continuous nature (mechanical, hydraulic, electrical) of physical systems. One of the earliest examples is the so-called Antikythera mechanism (150-100 BC), a model of the solar system that could predict astronomical positions [16]. Its origin remains uncertain, but more intriguing is the fact that no other computational device of similar complexity was discovered until thousand years later. Other analog computers were used for astronomical, navigational, and engineering purposes, e.g., astrolabes (attributed to Hipparcus, c. 190 – c. 120 BC), slide rules (1620-1630), planimeters (1814), and more recently, for simulating the dynamics of economic systems [17] or as air-missile controllers calculating ballistic trajectories [18].

Charles Babbage (1791-1871) designed the Difference Engine and the Analytical Engine, two early mechanical devices closely related to programmable computers [19]. The latter was an ambitious design embedding the transition from specialized calculating devices (hardware), to general-purpose computers performing a series of operations (software) described by a programmer [20]). Almost a hundred years later, Alan Turing beautifully recapitulated the hardware-software separation using an"universal computer", i.e., an abstract machine that can simulate any complex function using simple rules [21]. This mathematical breakthrough had a lasting impact in many domains, including artificial intelligence [22], cognitive psychology [23], neuroscience [24], linguistics [25] and physics [26], spurring the construction of the first electronic computers for military and scientific purposes in the 1940s and their subsequent descendants in administrative domains during the 1960s.

An important preadaptation to the success of digital computers lies in the development of binary encoding, which has its roots in Sanskrit poetic metres by Acharya Pingala. Functionally equivalent to an analog computer, a digital computer stores information in a discrete manner, using mechanical or electrical states of memory devices. Babbage's Analytical Engine incorporated digital arithmetic as well as many of the elements found in modern computers, but he could not possibly anticipate two important benefits of electronics [27]. First, electronic computers can operate much faster than mechanical computers. And second, electronics brought an exponential reduction of production costs with every generation of the computer (see below). These advances paved the way to efficient and smaller machines, and thus triggering the massive adoption of the "personal computer" [28, 29] and new types of human-machine interactions [30] in the 1970s.

Still, the choice of physical substrate cannot fully explain the evolution of computing [31]. Competitive and cooperative interactions also triggered the emergence of innovations in computing. In biology, new types of interactions, e.g., new modes of cell-cell attachment during development, can trigger major transitions in multicellular identity and complexification [32]. Similarly, interactions between computers and humans (and more recently among computers themselves) spurred the growth of software and hardware. An expected consequence (advanced by the psychologist J. C. R. Licklider in 1960 [33]) was the establishment of a symbiotic relationship between portable computing devices (e.g., smartphones) and humans [34], with deep implications in cognition [35], neuroscience [36], and ethics [37, 38].

Finally, diversity of computing niches has been greatly expanded by the deployment of the internet. Electronic interactions mirror social behaviours evolved over millions of years [39], providing new tools for the study of social systems [40]) and facilitated social cooperation at very-large spatial and temporal scales. For example, in the establishment of open-source software communities and online social networks [41], but also as a key ingredient in spreading the risks associated to information technologies [42].

3 Cultural evolution of computing

Information technologies are supported by many other knowledge domains that evolved through a cumulative cultural process. At the same time, computing affects the tempo and mode of cultural evolution, greatly accelerating innovation processes driven by the recombination of present technologies (see Figure 1a). We can recapitulate the evolution of computing as the emergence of new entities (material culture, algorithms) interacting synergistically with humans to accelerate the rate of cultural and technological change (see Figure 1c). This co-evolution between culture and computing should leave a signature in the rich "fossil" record of information technologies, but we cannot make testable predictions of technological trajectories without detailed information about the underlying evolutionary rules. This makes very difficult (or even impossible) to assess whenever well-known trends like Moore's law could be maintained in the future. Of particular interest here are the existence of large-scale trends in software evolution and how software complexity has been driven by common development mechanisms, i.e., social transmission, tinkering, optimization. These questions are a natural target for cultural evolution research, which has studied related topics in other domains.

3.1 Acceleration of cultural change

A main difference between computing and other technologies, like stone tools, is the exponential rate of change of the former (see Figure 1c) [5]. Achaulean handaxe technology remained stable over hundreds of thousands of years [43] while the number of transistors in integrated circuits is growing exponentially, doubling about every two years (Moore's law, see [44]). Although the predicted growth of hardware performance and storage capacity has been verified in the last six decades (see shaded area in Figure 1b), technological progress is ultimately bounded by physical constrains [45] as well as the capacity to successfully accumulate production experience [46]. This raises concerns about the future growth of computing. A systematic study shows Moore's law is not universal across all technological domains [47]. Instead, the cultural dynamics of artifacts, organisms and inventions is predicted to follow a characteristic S-curve trajectory consisting of an initial phase of slow growth followed by a rapid ascent that eventually saturates due to constrains. Multiple limiting factors, including strong selection, complexity barriers [48], population bottlenecks [49], and excess of imitation [50], can contribute to the stagnation of cultural diversity.

Exponential growth of redundant information is a mounting problem for information technology. Uncontrolled wasting of resources is particularly noticeable in the field of software development. For example, computer scientist Niklaus Wirth observed that "software speed is decreasing more quickly than hardware speed is increasing" (Wirth's law) [51], i.e., software ever-grows hardware. This phenomenon has even received a name, "software bloating", which describes the perception that the system is becoming slower, or demands more disk space or processing power with each new version. The causes of this unwarranted software complexity are not fully understood, but they seem related to features of information technology. Software in the 1960s and 1970's was severely constrained by memory and performance limitations, and programmers spent a lot of time optimizing their software. Indeed, the capacity for developing efficient software was one of the most valuable traits for programmers. This situation was reversed by the exponential growth of computing power and its much-reduced costs.

The theory of cultural evolution could be used to understand why software bloating occurs (and perhaps stop it). Evolution of complex technology depends on the gradual accumulation of many small changes over time, a process known as "ratcheting" [52]. Without reliable social transmission (e.g., high-fidelity copying) many cultural traits are short-lived, and insufficient diversity can prevent the selection of complex adaptations [53] (but see [54]). Low transmission fidelity (and the absence of cumulative culture) has been proposed as an explanation for the limited evidence of complex artefacts in non-human animals [55]. On the other hand, an excess of social transmission is non-adaptive. Digital substrate allows high-fidelity transmission "for free", and thus suggesting the importance of proper filtering mechanisms [56].

3.2 Technological Networks

Culture is a complex dynamic system involving multiple causal levels. However, many theoretical models map full systems onto a few traits or elements, perhaps those playing a key role in the system. This simplified representation, although useful, limits model's applicability and does not take into account indirect effects caused by multiple interactions between elements. Fully understanding how cultural evolution proceeds on (and regulates) these interactions necessitates a systems approach [57]. Hardware and software systems are no different, and they can be described as network of interconnected elements, e.g., an electronic circuit. Similarly, a software system, e.g., an operating system, a word-processor or a video game, consists of a large number of code parts, e.g., subroutines or objects, interacting with each other in a network. Optimality principles dictate how software systems must be arranged, but it is very difficult to avoid the (often non-obvious) design barriers imposed by complexity [58]. Evolutionary constrains are also evidenced by common patterns in the internal structure of software, that is, structural heterogeneity [59, 60, 61, 62], small-world behaviour [63], motif distribution [64], and modular organisation [63, 65, 66, 67]. This set of topological features is shared by many evolving systems [68, 69], and thus likely to be displayed by other technological and cultural systems. An explanation for this universal behaviour resides in the ubiquity of tinkering mechanisms [70]. In particular, simple evolutionary models based on copying and random mutations predict that the frequency distribution of connections attached to any node must be highly heterogeneous (see Figure 2a).

Demography also mediates technological complexity. The "treadmill model" proposes that group size is a main causal factor for the evolution of cultural complexity [49]. A large group can support technological specialists from whom others can learn. This hypothesis has been contested on several empirical and theoretical fronts [71]. For example, it has been suggested that available evidence does not always support the predictions of the model [72]. From a broader perspective, population size is not the only determinant of cultural complexity, which can also be influenced by exogenous factors. Moreover, population size is too coarse for accurately evaluating hypotheses about demographic effects on culture and technology. A more realistic representation of social interactions is needed, that is, we need to understand how properties of social networks (e.g., connectivity, heterogeneity, modular organisation) drive cultural diversity and complexity [73].

Software communities offer an excellent case study for determining the demography-complexity relationship. Highly-detailed records of both social interactions and evolved cultural artifacts, i.e., a software system, are available. An analysis of e-mail exchanges in distributed communities of open-source software projects shows that communication load is not evenly distributed (as it will be expected in a leaderless group). Open-source software communities consist of a small team of core programmers taking care of the majority of work, while the rest of the community plays a supporting role by detecting and reporting software errors ("bugs" [74]). This hierarchical, two-layered organisation pattern can be detected by measuring the rich-club coefficient $\Phi(k)$ or the extend to which well-connected nodes relate to each other (see Figure 2a) [75]. The relatively small size of the core team reflects prestige and reputation biases [76] as well as limiting communication overheads, as suggested by the Brook's law "adding more people to a late software project makes it later" [77]. Interestingly, these observations are consistent with a recent cultural study suggesting how group size depends on the complexity of the task [78].

3.3 Digital worlds

As a result of information technologies and the advent of a digital economy, new media and in particular several "digital worlds" have emerged, providing a solid test bed for cultural evolution. These virtual worlds are becoming increasingly more important in our lives [79] and provide several interesting qualities for research. These range from complete "genotype" and "phenotype" mapping and the ability to sometimes engineer experiments instead of simply analyzing historical trends. A great example of these features can be found in video game data, from social and wealth dynamics in massive multiplayer games [80, 81], virtual epidemics [82] to characterizing individual learning and social transmission in speed running communities [83].

In contrast to non-digital technologies, direct estimation of fitness or utility is readily available, whether though sales or number of users, allowing for models of technological succession and cultural evolution involving complete descriptions of the underlying fitness landscape. Despite the positive features these systems offer, digital worlds also provide their own unique set of challenges. Namely, there is an ongoing realization of the dangers of link rot in digital archaeology [84]. Information in the form of internet hyperlinks and documents is in a process of constant decay and great effort must be put into conservation efforts of historical data [85]. Only certain domains are well preserved, typically where there are a centralised authorities distributing and controlling the information, such as scientific publications and patents. However, even in those areas documents often reference non canonical sources like news posts to establish meaning and context, which eventually disappear.

4 Major transitions in information technology

Application of evolutionary theory to investigate human cultural diversity has a long tradition in anthropology [86]. In the biological domain, Maynard-Smith and Száthmary defined singular events in the history of our biosphere in which the canonical understanding of evolutionary theory breaks [87]. These singularities arise when there is a discontinuity in the very rules underlying the evolutionary process. For instance, some transitions involve the creation of a new individual, upon which selection can also apply, requiring an expansion of evolutionary theory to include multi-level selection [88]. This new unit of selection can be loosely distributed in a form of society through collective behaviour [89, 90] or more tightly knit in a single compact body [91, 92]. Another type of transition comes from qualitative changes in the manner in which biological information is stored, encoded or transmitted. For instance, the evolution of a separate system for storing and executing functions (the DNA-protein function divide) from a world where RNA would carry out both functions [93, 94] (in this picture the ribosome is an intermediary form of the eventual separation of functions, composed of both rRNA and aminoacids). The evolution of sex as a mode of producing genetic variation by recombination of information from two parents [95] is typically classified in this category of information related transitions. Finally, another class of major transitions is found in the advent of language, cognition and memory, which allow for non-genetic information to be stored and reliably transmitted between individuals [96], i.e. the building blocks required for cumulative culture. In specific case of human culture, it has also been recently proposed that a separate domain consisting of evolving cultural traits (called the 'sociont' [97]) coexists with the human biological substrate undergoing its own evolutionary processes. This provides the basis for a transition in individuality [98], where cultures would be able to evolve and propagate in a purely cognitive medium given by human societies.

We take the framework of major transitions to reexamine the evolution of information technologies and computation [99]. In particular, we can see the discontinuities in information technologies [100] as radical departures from the prior set of evolutionary rules operating within the technological space of computing processes. It is important to realise that although new computational paradigms might emerge, each major transition often did not erase the prior mechanisms of computation. Similarly to the process of evolving new writing implements, the invention of the printing press did not make quills and chisels obsolete (see Figure 1c). Returning to the biological parallelisms, the coexistence of varied and overlapping strategies in systems undergoing major transitions is indicative of symbiotic processes or specialist strategies. Tools and species can become better suited for specific niches that allow them to coexist, sometimes forming strong cross-feeding interactions like syntrophy [101]. This is the case of the hardware-software pair, which have co-evolved from a single entity in the advent of computing technologies.

4.1 Hardware

Both software and hardware exist in a dual state, their fitness evaluated from their individual and paired characteristics. The hardware-software symbiont evolution follows strong social dynamics but is also majorly driven by an extrinsic fitness component derived from energetic and temporal constrains. In the case of hardware, this translates to several universal and continuous laws such as Moore's law and s-curves mentioned prior, which establish the overall evolutionary trends to be expected in terms of computing power and the economy of computation. As the material component of information technologies, hardware innovations allow for larger and more complex algorithms (software bloating) to be run, which in turn can drive the demand for increasingly more powerful hardware.

Several key transitions in hardware evolution can be identified regarding to the nature of information and computation. At first, computers were physical devices operating by means of turning gears and pulleys and relaying the results of the computation by macroscopic mechanical processes. In this mechanical computer stage hardware and software were a single entity, often the computing apparatus could not be reconfigured to do other tasks (like the Antikythera mechanism [102]) or in the case of the difference engine, changing the computational process required rearrangement of physical elements. This first transition thus, is the divergence of hardware software as distinct entities, similarly to the division of labor in the RNA world discussed before [103]. It wasn't until the creation of EDVAC that software came into being [104], freeing hardware from the constrains of being able to be physically reconfigured and allowing it to explore new evolutionary paths.

Another major transition took place in the switch from analogical to digital computers. This change in information format entailed several hardware developments, including storage devices such as punched cards, vacuum tubes and transistors, but also on the methods of processing of digital information. The biological analogies here fall short, as most genetic information (RNA, DNA and epigenetic information in methylation) is digital by nature [96], however this does not mean that analog genetic information could not devised in synthetic systems such as protocells or that analog information was not present in the early stages of life [105, 106].

After the hardware-software separation, the Von Neumann architecture became an established framework that has persisted until the present. Most improvements have come from the consistent development of new memory units and processing components. The first transition happened in the 1950s with the replacing of vacuum tubes with transistors, which had much smaller footprint, consumed much less energy and gave less heat, and thus, were much more reliable than their electromechanical counterparts [104]. The next major innovation after the digital electronic computer was the invention of integrated circuit computers, made of printable wafers that connected individual computing elements, which allowed for further miniaturization and mass production of computers. Figure 2A bottom shows a characteristic scaling in planar integrated circuits: Rentian scaling [107]. This rule states that in a given square section of an integrated circuit, the number of wires crossing the boundary and the number of gates / computing elements inside of it are related in a mathematically predictable way that depends on the type of circuit (different colours in the figure). In the biological domain, similar relations have been found in other structures relating to major transitions: neuron topological organization [108], which hints at deeper connections between computing substrates that have a physical embedding and spatial constrains [109].

Currently, the path to increase computational power seems to rest on increased parallelism [110], using multi-core processors and massive GPUs. The development of a hierarchy of form that spans multiple scales (Figure 2a) is reminiscent to McShea's theory on the evolution of complexity and major evolutionary transitions in our biosphere [111]. He argues that a hierarchy of components of segregated scales naturally emerges in the biological domain (proteins interacting in compartments, which make cells of different types, which in turn are organized in organs of different kinds) purely by the necessity to diversify and the existence of incompatibilities at the lower level. By allowing the segregation of incompatible components it is possible to continuously increase the complexity and insulate working sets of genes in the form of toolkits.

4.2 Software

As discussed above, the first major transition relating to software is its emergence in the design of "stored-program" computers during the development of EDVAC [104]. In the 1940s, most applications were written in machine language: a direct instruction to the machine with no human abstraction. Programming in machine language was a time-consuming effort: simple algorithmic processes, like multiplication, required many steps in machine-level instructions. In addition, there was a high risk of duplicating efforts since different users had their own recipes for the same processes. This situation can be partially alleviated by replacing machine codes with "high-level" languages (meaning a language that affords some level of abstraction in the code instructions and is more lenient to human coders) like Plankalkül, one of the first proposed by Konrad Zuse in 1943 [112]. As an emerging symbolic technology, this first epoch after the creation of software is characterized by extremely innovative approaches leading to the main evolutionary branches observed today [113], rooted in Lisp and Fortran (see Figure 2C). Very much like the Cambrian explosion [114, 115], when a sudden burst in new body plan designs appear after a relative period of stasis [116], the following decades mark the first appearance of core features of higher-level languages that any modern programmer would recognize, such as object-oriented programming (see below), logic programming, one-pass compiling and query languages.

Another aspect worth considering beyond feature evolution is the rate of language creation and the amount of influence a language has in future generations and receives from its past peers. This amounts to establishing some sort of inheritance in a domain where influence can be established at any distance in space or time. In biology, a single uninterrupted thread of inheritance of genetic material exists between every natural living being on earth and the last universal common ancestor [117]. But in the cultural domain, parenthood is not restricted to any fixed number of technologies in the current generation or even living technologies for that matter. In the case of programming languages, a striking pattern in rates of recombination or number of parent technologies has been recently found (Figure 2B [113]) which is reminiscent of another major transition in evolution: the evolution of sexual reproduction. Specifically, a discontinuity in the rates of inheritance but not in the rates of creation can be observed around the year 1983 in a pattern that otherwise holds for decades. This sudden increase in recombination has been linked to the emergence of personal computing [113] and is thus linked to hardware innovations that prompted changes in social interactions and culture transmission. Sex in the biological domain has been associated to multiple benefits: from creating meaningful variation in variable environments to parasite avoidance [118]. In the case of programming languages, it remains to be determined whether there were any external pressures driving this transition beyond the systemic changes to culture transmission mentioned before.

The final major transition takes place when new levels of organization appear and programs become components of larger code structures such as operating systems and applications (see Figure 2A). A specially interesting example is open software libraries, where hundreds of programmers are required to transfer knowledge and functionality to other elements of the global structure without negatively interfering with code that they themselves cannot not know before hand. This is particularly significant given that most code is produced by tinkering [119, 70], i.e. programmers copy lines of code that already exist and rewrite some of the components to adapt its functionality, but can incur in serious danger of negatively interfering with the original code and function. Analogous to the transition to multicellular life, this adaptation has produced similar solutions than those observed in biology. For example, in order to be able to create complex multicellular bodies with ample functionality, compartmentalization of functions is key to avoid negative epistasis [120, 111]. Toolkits of genes specialized in creating biological form and regulating compartment interactions are of utmost importance in this picture of the transition to multicellular life [121] and mirror the process of establishing subroutines with controllable scope, protocols for communication as well as various programming standards that can enhance component re-usability [122]. In the 1960s, Alan Kay embedded these biologically-inspired design principles in the programming language Smalltalk [123]. Smalltalk design (typified as “object-oriented programming”) encourages programmers to think of software as a network of “biological cells and/or individual computers only able to communicate with messages” and this principle has been widely adopted by many modern languages, e.g., C++, Java and Python.

5 Discussion

We have briefly reviewed the history of computation and provided two alternative perspectives on its evolution. First, a more standard view using cumulative cultural evolution and tools provided by complexity theory and network science. Such view has found ample success in modeling and establishing explanatory causality of many phenomena in the cultural domain, from rates of cultural change and technological succession to community structure and evolution in technological networks. We proposed this framework will be able to make ample use of the wealth of data provided by digital worlds, social media and our current digital-oriented economy. Secondly, we explored an alternative framework based the major transitions of evolution as studied in the biological sciences. This perspective is particularly valuable in evolutionary theory as it has allowed researchers from diverging fields to integrate distinct phenomena that yield discontinuous, punctuated evolutionary dynamics with a cohesive body of theory. Such approach can offer novel insights into the key ingredients and agents responsible for technological revolutions and should be combined with the previous approach to tackle both continuous and discontinuous evolution in the technological domain.

Many open questions remain in this area that would benefit from a comprehensive approach that incorporates the lessons learned in biological evolutionary theory. For instance, it is still unclear what the role of demographics in cultural complexity evolution is, as evidenced by the ongoing discussion on the Tasmanian culture collapse [72, 124]. The effect of population size has been specifically addressed in computing technologies by the eponymous Brook's law, which implies a finite optimal size in culture production teams. But such effects can be muddled by other explanatory variables including copying fidelity [53, 54], community structure [125] and social learning rates [50, 126]. Untangling these factors requires additional efforts in designing experimental setups with controllable environments, instead of relying on purely observational data. Such theory would prove invaluable in establishing optimal culture production rates in science and beyond, affording us a chance at overcoming redundancy, technological bloating and cultural collapse.

Predicting the uses of computing devices and technologies has proven to be an ambitious but ultimately unfruitful effort. From military and academic uses in its infancy, information technologies have been exapted into various domains, including labour and social endeavors, and now permeates almost every aspect of our lives. However, there are some clues towards where information technologies and their paradigms will spread next. One of such areas is synthetic biology, where there is an unambiguous motivation to translate the lessons learned from modular and scalable software production into engineering living machines [127, 128]. The influence of electronics is clearly tangible in the use of language: logic gates, transfer functions and truth tables have become common parlance in the community. The goals are straightforward: to imitate the enormous success of hardware and software engineering in the development of "wetware", a vision that has its roots in the cybernetics of the last century [129, 130, 131].

The cross-fertilization between various human endeavours and computing has provided apt new insights [132, 129] and undoubtedly accelerated the rates cultural evolution [4, 5]. However, it has also had the concerning effects of reducing the value of human labour [133], decoupling population size from economic growth [47]. Advanced computational tools like deep learning now threaten to supersede the need for human workers in many fields from transportation to retail [134]. Additionally, it poses the serious threat of translating something that is yet poorly understood, inheriting flawed frameworks into new domains. For instance, the impact of information technologies on social media is still a source of controversy [135, 42]. Some authors contend that the current juncture in political polarization and misinformation is indeed boosted by the hyper-connectivity of social media and the use of content presentation algorithms that promote echo chambers of discourse. Addressing these concerns and understanding the potential repercussions of this class of phenomena should be a priority for the field before carrying over the cultural baggage of computing to other domains.

6 Acknowledgments

SDN is supported by Beatriu de Pinós grant 2019-BP-00206, program of the AGAUR Generalitat de Catalunya. SV is supported by the Spanish Ministry of Science and Innovation through the State Research Agency (AEI), grant PID2020-117822GB-I00/AEI/10.13039/501100011033 and through the Spanish National Research Council (CSIC), grant CONEXIONES-Vida/202120E47. This review is dedicated to the memory of Sir Clive Sinclair.

References

[1] Daniela Cerqui. From turing to the information society. In *Alan Turing: Life and Legacy of a Great Thinker*, pages 59–74. Springer, 2004.

[2] Matjaž Gams and Tine Kolenik. Relations between electronics, artificial intelligence and information society through information society rules. *Electronics*, 10(4):514, 2021.

[3] Thomas Haigh. The history of information technology. *Annual review of information science and technology*, 45(1):431–487, 2011.

[4] Martin Hilbert and Priscila López. The world's technological capacity to store, communicate, and compute information. *science*, 332(6025):60–65, 2011.

[5] R Alexander Bentley and Michael J O'Brien. *The acceleration of cultural change: From ancestors to algorithms.* MIT Press, 2017.

[6] Claudia Wagner, Markus Strohmaier, Alexandra Olteanu, Emre Kıcıman, Noshir Contractor, and Tina Eliassi-Rad. Measuring algorithmically infused societies. *Nature*, pages 1–6, 2021.

[7] Peter B Campbell. The anthropocene, hyperobjects and the archaeology of the future past. *Antiquity*, pages 1–16, 2021.

[8] Ricard V Solé, Sergi Valverde, and Carlos Rodriguez-Caso. Convergent evolutionary paths in biological and technological networks. *Evolution: Education and Outreach*, 4(3):415–426, 2011.

[9] John Henry Holland et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[10] John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4(2):87–112, 1994.

[11] Lorenzo Cecutti, Anthony Chemero, and Spike WS Lee. Technology may change cognition without necessarily harming it. *Nature Human Behaviour*, pages 1–3, 2021.

[12] Ton Sales. Llull as computer scientist or why llull was one of us. In *International AMAST Workshop on Aspects of Real-Time Systems and Concurrent and Distributed Software*, pages 15–21. Springer, 1997.

[13] Martin Campbell-Kelly, William Aspray, Daniel P Snowman, Susan R McKay, and Wolfgang Christian. Computer a history of the information machine. *Computers in Physics*, 11(3):256–257, 1997.

[14] Thomas Haigh and Paul E Ceruzzi. *A New History of Modern Computing*. MIT Press, 2021.

[15] Jeremy G Butler. A history of information technology and systems. *University of Arizona*, 2012.

[16] Tony Freeth, David Higgon, Aris Dacanalis, Lindsay MacDonald, Myrto Georgakopoulou, and Adam Wojcik. A model of the cosmos in the ancient greek antikythera mechanism. *Scientific reports*, 11(1):1–15, 2021.

[17] Chris Bissell. Historical perspectives-the moniac a hydromechanical analog computer of the 1950s. *IEEE Control Systems Magazine*, 27(1):69–74, 2007.

[18] B Jack Copeland. The modern history of computing. 2000.

[19] Michael Lindgren. *Glory and failure: the difference engines of Johann Müller, Charles Babbage and Georg and Edvard Scheutz*. PhD thesis, Linköpings universitet, 1987.

[20] Eugene Eric Kim and Betty Alexandra Toole. Ada and the first computer. *Scientific American*, 280(5):76–81, 1999.

[21] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.

[22] Marvin Minsky and Seymour A Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

[23] Hilary Putnam. *Brains and behaviour*. Blackwell, 1968.

[24] David Marr. *Vision: a computational investigation into the human representation and processing of visual information*. San Francisco: Freeman, 1982.

[25] Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

[26] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[27] Claude E Shannon. A symbolic analysis of relay and switching circuits. *Electrical Engineering*, 57(12):713–723, 1938.

[28] Alan C Kay. A personal computer for children of all ages. In *Proceedings of the ACM annual conference-Volume 1*, 2011.

[29] Douglas K Smith and Robert C Alexander. *Fumbling the future: How Xerox invented, then ignored, the first personal computer.* iUniverse, 1999.

[30] Julie A Jacko. *Human computer interaction handbook: Fundamentals, evolving technologies, and emerging applications.* CRC press, 2012.

[31] Cristopher Moore and Stephan Mertens. *The nature of computation.* OUP Oxford, 2011.

[32] Stuart A Newman. Inherent forms and the evolution of evolution. *Journal of Experimental Zoology Part B: Molecular and Developmental Evolution*, 332(8):331–338, 2019.

[33] Joseph CR Licklider. Man-computer symbiosis. *IRE transactions on human factors in electronics*, (1):4–11, 1960.

[34] Rich Ling, Leopoldina Fortunati, Gerard Goggin, Sun Sun Lim, and Yuling Li. *The Oxford Handbook of Mobile Communication and Society.* Oxford University Press, 2020.

[35] Henry H Wilmer, Lauren E Sherman, and Jason M Chein. Smartphones and cognition: A review of research exploring the links between mobile technology habits and cognitive functioning. *Frontiers in psychology*, 8:605, 2017.

[36] Eveline A Crone and Elly A Konijn. Media use and brain development during adolescence. *Nature communications*, 9(1):1–10, 2018.

[37] Geoffrey Miller. The smartphone psychology manifesto. *Perspectives on psychological science*, 7(3):221–237, 2012.

[38] Frantz Rowe. Contact tracing apps and values dilemmas: A privacy paradox in a neo-liberal world. *International Journal of Information Management*, 55:102178, 2020.

[39] Bruno Gonçalves, Nicola Perra, and Alessandro Vespignani. Modeling users' activity on twitter networks: Validation of dunbar's number. *PloS one*, 6(8):e22656, 2011.

[40] Jake M Hofman, Duncan J Watts, Susan Athey, Filiz Garip, Thomas L Griffiths, Jon Kleinberg, Helen Margetts, Sendhil Mullainathan, Matthew J Salganik, Simine Vazire, et al. Integrating explanation and prediction in computational social science. *Nature*, pages 1–8, 2021.

[41] Valerio Arnaboldi, Andrea Passarella, Marco Conti, and Robin IM Dunbar. *Online social networks: human cognitive constraints in Facebook and Twitter personal graphs.* Elsevier, 2015.

[42] Joseph B Bak-Coleman, Mark Alfano, Wolfram Barfuss, Carl T Bergstrom, Miguel A Centeno, Iain D Couzin, Jonathan F Donges, Mirta Galesic, Andrew S Gersick, Jennifer Jacquet, et al. Stewardship of global collective behavior. *Proceedings of the National Academy of Sciences*, 118(27), 2021.

[43] Meir Finkel and Ran Barkai. The acheulean handaxe technological persistence: A case of preferred cultural conservatism? In *Proceedings of the Prehistoric Society*, volume 84, pages 1–19. Cambridge University Press, 2018.

[44] Gordon E Moore et al. Cramming more components onto integrated circuits, 1965.

[45] M Mitchell Waldrop. The chips are down for moore's law. *Nature News*, 530(7589):144, 2016.

[46] J Doyne Farmer and François Lafond. How predictable is technological progress? *Research Policy*, 45(3):647–665, 2016.

[47] Vaclav Smil. *Growth: from microorganisms to megacities.* Mit Press, 2019.

[48] James McNerney, J Doyne Farmer, Sidney Redner, and Jessika E Trancik. Role of design complexity in technology improvement. *Proceedings of the National Academy of Sciences*, 108(22):9008–9013, 2011.

[49] Joseph Henrich. Demography and cultural evolution: how adaptive cultural processes can produce maladaptive losses—the tasmanian case. *American antiquity*, 69(2):197–214, 2004.

[50] Salva Duran-Nebreda and Sergi Valverde. Imitation-driven cultural collapse. 2021.

[51] Niklaus Wirth. A plea for lean software. *Computer*, 28(2):64–68, 1995.

[52] Claudio Tennie, Josep Call, and Michael Tomasello. Ratcheting up the ratchet: on the evolution of cumulative culture. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1528):2405–2415, 2009.

[53] Hannah M Lewis and Kevin N Laland. Transmission fidelity is the key to the build-up of cumulative culture. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1599):2171–2180, 2012.

[54] Carmen Saldana, Joël Fagot, Simon Kirby, Kenny Smith, and Nicolas Claidière. High-fidelity copying is not necessarily the key to cumulative cultural evolution: a study in monkeys and children. *Proceedings of the Royal Society B*, 286(1904):20190729, 2019.

[55] Claudio Tennie, Lydia M Hopper, and Carel P van Schaik. On the origin of cumulative culture: Consideration of the role of copying in culture-dependent traits and a reappraisal of the zone of latent solutions hypothesis. In *Chimpanzees in Context*, pages 428–453. University of Chicago Press, 2020.

[56] Rachel L Kendal, Neeltje J Boogert, Luke Rendell, Kevin N Laland, Mike Webster, and Patricia L Jones. Social learning strategies: Bridge-building between fields. *Trends in cognitive sciences*, 22(7):651–665, 2018.

[57] Andrew Buskell, Magnus Enquist, and Fredrik Jansson. A systems approach to cultural evolution. *Palgrave Communications*, 5(1):1–15, 2019.

[58] James Winters. Escaping optimization traps: the role of cultural adaptation and cultural exaptation in facilitating open-ended cumulative dynamics. *Palgrave Communications*, 5(1):1–13, 2019.

[59] Sergi Valverde, R Ferrer Cancho, and Richard V Sole. Scale-free networks from optimal design. *EPL (Europhysics Letters)*, 60(4):512, 2002.

[60] Giulio Concas, Michele Marchesi, Sandro Pinna, and Nicola Serra. Power-laws in a large object-oriented software system. *IEEE Transactions on Software Engineering*, 33(10):687–708, 2007.

[61] Panagiotis Louridas, Diomidis Spinellis, and Vasileios Vlachos. Power laws in software. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 18(1):1–26, 2008.

[62] Tin Yau Pang and Sergei Maslov. Universal distribution of component frequencies in biological and technological systems. *Proceedings of the National Academy of Sciences*, 110(15):6235–6239, 2013.

[63] Sergi Valverde and Ricard V Solé. Hierarchical small worlds in software architecture. *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms (S6 (Special Issue on Software Engineering and Complex Networks))*, 14:1, 2007.

[64] Sergi Valverde and Ricard V Solé. Network motifs in computational graphs: A case study in software architecture. *Physical Review E*, 72(2):026107, 2005.

[65] Miguel A Fortuna, Juan A Bonachela, and Simon A Levin. Evolution of a modular software network. *Proceedings of the National Academy of Sciences*, 108(50):19985–19989, 2011.

[66] Hui Li, Hai Zhao, Wei Cai, Jiu-Qiang Xu, and Jun Ai. A modular attachment mechanism for software network evolution. *Physica A: Statistical Mechanics and its Applications*, 392(9):2025–2037, 2013.

[67] Pablo Villegas, Miguel A Muñoz, and Juan A Bonachela. Evolution in the debian gnu/linux software network: analogies and differences with gene regulatory networks. *Journal of the Royal Society Interface*, 17(163):20190845, 2020.

[68] Ricard V Solé, Ramon Ferrer-Cancho, Jose M Montoya, and Sergi Valverde. Selection, tinkering, and emergence in complex networks. *Complexity*, 8(1):20–33, 2002.

[69] Petr Keil, AAM MacDonald, Kelly S Ramirez, Joanne M Bennett, Gabriel E García-Peña, Benjamin Yguel, Bérenger Bourgeois, and Carsten Meyer. Macroecological and macroevolutionary patterns emerge in the universe of gnu/linux operating systems. *Ecography*, 41(11):1788–1800, 2018.

[70] Ricard Solé and Sergi Valverde. Evolving complexity: how tinkering shapes cells, software and ecological networks. *Philosophical Transactions of the Royal Society B*, 375(1796):20190325, 2020.

[71] Claes Andersson and Dwight Read. The evolution of cultural complexity: Not by the treadmill alone. *Current Anthropology*, 57(3):261–286, 2016.

[72] Krist Vaesen, Mark Collard, Richard Cosgrove, and Wil Roebroeks. Population size does not explain past changes in cultural complexity. *Proceedings of the National Academy of Sciences*, 113(16):E2241–E2247, 2016.

[73] Dominik Deffner, Anne Kandler, and Laurel Fogarty. Effective population size for culturally evolving traits. *bioRxiv*, 2021.

[74] Fred R Shapiro. Etymology of the computer bug: History and folklore. *American Speech*, 62(4):376–378, 1987.

[75] Sergi Valverde and Ricard V Solé. Self-organization versus hierarchy in open-source social networks. *Physical Review E*, 76(4):046118, 2007.

[76] Jungpil Hahn, Jae Yun Moon, and Chen Zhang. Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. *Information Systems Research*, 19(3):369–391, 2008.

[77] Frederick P Brooks Jr. *The mythical man-month: essays on software engineering.* Pearson Education, 1995.

[78] Abdullah Almaatouq, Mohammed Alsobay, Ming Yin, and Duncan J Watts. Task complexity moderates group synergy. *Proceedings of the National Academy of Sciences*, 118(36), 2021.

[79] Andreas M Kaplan and Michael Haenlein. The fairyland of second life: Virtual social worlds and how to use them. *Business horizons*, 52(6):563–572, 2009.

[80] Michael Szell and Stefan Thurner. Measuring social dynamics in a massive multiplayer online game. *Social networks*, 32(4):313–329, 2010.

[81] Benedikt Fuchs and Stefan Thurner. Behavioral and network origins of wealth inequality: Insights from a virtual world. *PloS one*, 9(8):e103503, 2014.

[82] Eric T Lofgren and Nina H Fefferman. The untapped potential of virtual game worlds to shed light on real world epidemics. *The Lancet infectious diseases*, 7(9):625–629, 2007.

[83] Manuel Lafond. The complexity of speedrunning video games. In *9th International Conference on Fun with Algorithms (FUN 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[84] John Markwell and David W Brooks. "link rot" limits the usefulness of web-based educational materials in biochemistry and molecular biology. *Biochemistry and Molecular Biology Education*, 31(1):69–72, 2003.

[85] Jonathan Zittrain, Kendra Albert, and Lawrence Lessig. Perma: scoping and addressing the problem of link and reference rot in legal citations. *Harv. L. Rev. F.*, 127:176, 2013.

[86] Robert L Carneiro. *Evolutionism in cultural anthropology: a critical history.* Routledge, 2018.

[87] John Maynard Smith and Eors Szathmary. *The major transitions in evolution.* Oxford University Press, 1997.

[88] Samir Okasha. *Evolution and the Levels of Selection.* Oxford University Press, 2006.

[89] Bert Hölldobler, Edward O Wilson, et al. *The ants.* Harvard University Press, 1990.

[90] Ricard Solé, Daniel R Amor, Salva Duran-Nebreda, Núria Conde-Pueyo, Max Carbonell-Ballestero, and Raul Montañez. Synthetic collective intelligence. *Biosystems*, 148:47–61, 2016.

[91] William C Ratcliff, R Ford Denison, Mark Borrello, and Michael Travisano. Experimental evolution of multicellularity. *Proceedings of the National Academy of Sciences*, 109(5):1595–1600, 2012.

[92] Salva Duran-Nebreda, Adriano Bonforti, Raúl Montanez, Sergi Valverde, and Ricard Solé. Emergence of proto-organisms from bistable stochastic differentiation and adhesion. *Journal of The Royal Society Interface*, 13(117):20160108, 2016.

[93] Alexander Rich. On the problems of evolution and biochemical information transfer. *Horizons in biochemistry*, pages 103–126, 1962.

[94] Marc Neveu, Hyo-Joong Kim, and Steven A Benner. The "strong" rna world hypothesis: Fifty years old. *Astrobiology*, 13(4):391–403, 2013.

[95] Mauro Santos, Elias Zintzaras, and Eörs Szathmáry. Origin of sex revisited. *Origins of Life and Evolution of the Biosphere*, 33(4):405–432, 2003.

[96] Eva Jablonka and Marion J Lamb. *Evolution in four dimensions, revised edition: Genetic, epigenetic, behavioral, and symbolic variation in the history of life*. MIT press, 2014.

[97] Claes Andersson and Petter Törnberg. Toward a macroevolutionary theory of human evolution: The social protocell. *Biological Theory*, 14(2):86–102, 2019.

[98] Dinah R Davison, Claes Andersson, Richard E Michod, and Steven L Kuhn. Did human culture emerge in a cultural evolutionary transition in individuality? *Biological Theory*, pages 1–24, 2021.

[99] Sergi Valverde. Major transitions in information technology. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1701):20150450, 2016.

[100] R Alexander Bentley and Michael John O'Brien. Cultural evolutionary tipping points in the storage and transmission of information. *Frontiers in Psychology*, 3:569, 2012.

[101] Eric Libby and William C Ratcliff. Lichens and microbial syntrophies offer models for an interdependent route to multicellularity. *The Lichenologist*, 53(4):283–290, 2021.

[102] Tony Freeth, Yanis Bitsakis, Xenophon Moussas, John H Seiradakis, Agamemnon Tselikas, Helen Mangou, Mary Zafeiropoulou, Roger Hadland, David Bate, Andrew Ramsey, et al. Decoding the ancient greek astronomical calculator known as the antikythera mechanism. *Nature*, 444(7119):587–591, 2006.

[103] Freeman J Dyson. A model for the origin of life. *Journal of Molecular Evolution*, 18(5):344–350, 1982.

[104] Anthony JG Hey, Tony Hey, and Gyuri Pápay. *The computing universe: a journey through a revolution*. Cambridge University Press, 2014.

[105] Katarzyna Adamala and Jack W Szostak. Nonenzymatic template-directed rna synthesis inside model protocells. *Science*, 342(6162):1098–1100, 2013.

[106] Ricard V Solé, Andreea Munteanu, Carlos Rodriguez-Caso, and Javier Macía. Synthetic protocell biology: from reproduction to computation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1486):1727–1739, 2007.

[107] Bernard S Landman and Roy L Russo. On a pin versus block relationship for partitions of logic graphs. *IEEE Transactions on computers*, 100(12):1469–1479, 1971.

[108] Danielle S Bassett, Daniel L Greenfield, Andreas Meyer-Lindenberg, Daniel R Weinberger, Simon W Moore, and Edward T Bullmore. Efficient physical embedding of topologically complex information processing networks in brains and computer circuits. *PLoS computational biology*, 6(4):e1000748, 2010.

[109] Ed Bullmore and Olaf Sporns. The economy of brain network organization. *Nature Reviews Neuroscience*, 13(5):336–349, 2012.

[110] W Daniel Hillis. *The connection machine*. MIT press, 1989.

[111] Daniel W McShea. A complexity drain on cells in the evolution of multicellularity. *Evolution*, 56(3):441–452, 2002.

[112] Raúl Rojas, Cüneyt Göktekin, Gerald Friedland, and Mike Krüger. Plankalkül: The first high-level programming language and its implementation. 2000.

[113] Sergi Valverde and Ricard V Solé. Punctuated equilibrium in the large-scale evolution of programming languages. *Journal of The Royal Society Interface*, 12(107):20150249, 2015.

[114] S Conway Morris. Burgess shale faunas and the cambrian explosion. *Science*, 246(4928):339–346, 1989.

[115] Stephen Jay Gould. The evolution of life on the earth. *Scientific American*, 271(4):84–91, 1994.

[116] Niles Eldredge. Punctuated equilibria: an alternative to phyletic gradualism. In *Time frames*, pages 193–224. Princeton University Press, 2014.

[117] Eugene V Koonin. Comparative genomics, minimal gene-sets and the last universal common ancestor. *Nature Reviews Microbiology*, 1(2):127–136, 2003.

[118] John Maynard Smith and John Maynard-Smith. *The evolution of sex*, volume 4. Cambridge University Press Cambridge, 1978.

[119] François Jacob. Evolution and tinkering. *Science*, 196(4295):1161–1166, 1977.

[120] Enrique Flores and Antonia Herrero. Compartmentalized function through cell differentiation in filamentous cyanobacteria. *Nature Reviews Microbiology*, 8(1):39–50, 2010.

[121] Stuart A Newman and Ramray Bhat. Dynamical patterning modules: a" pattern language" for development and evolution of multicellular form. *International Journal of Developmental Biology*, 53(5-6):693–705, 2009.

[122] Sergi Valverde. The long and winding road: Accidents and tinkering in software standardization. 2021.

[123] Alan C Kay. The early history of smalltalk. *History of programming languages—II*, pages 511–598, 1996.

[124] Joseph Henrich, Robert Boyd, Maxime Derex, Michelle A Kline, Alex Mesoudi, Michael Muthukrishna, Adam T Powell, Stephen J Shennan, and Mark G Thomas. Understanding cumulative cultural evolution. *Proceedings of the National Academy of Sciences*, 113(44):E6724–E6725, 2016.

[125] Andrea B Migliano, Federico Battiston, Sylvain Viguier, Abigail E Page, Mark Dyble, Rodolph Schlaepfer, Daniel Smith, Leonora Astete, Marilyn Ngales, Jesus Gomez-Gardenes, et al. Hunter-gatherer multilevel sociality accelerates cumulative cultural evolution. *Science advances*, 6(9):eaax5913, 2020.

[126] Micael Ehn and Kevin Laland. Adaptive strategies for cumulative cultural learning. *Journal of theoretical biology*, 301:103–111, 2012.

[127] Ramiz Daniel, Jacob R Rubens, Rahul Sarpeshkar, and Timothy K Lu. Synthetic analog computation in living cells. *Nature*, 497(7451):619–623, 2013.

[128] Javier Macía, Francesc Posas, and Ricard V Solé. Distributed computation: the new wave of synthetic biology devices. *Trends in biotechnology*, 30(6):342–349, 2012.

[129] Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*. MIT press, 2019.

[130] D Ewen Cameron, Caleb J Bashor, and James J Collins. A brief history of synthetic biology. *Nature Reviews Microbiology*, 12(5):381–390, 2014.

[131] Víctor De Lorenzo and Antoine Danchin. Synthetic biology: discovering new worlds and new words: the new and not so new aspects of this emerging research field. *EMBO reports*, 9(9):822–827, 2008.

[132] John Von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. In *Automata Studies.(AM-34), Volume 34*, pages 43–98. Princeton University Press, 2016.

[133] Lanier Jaron. *You Are Not a Gadget: A Manifesto*. New York: Knopf, 2010.

[134] Joseph E Aoun. *Robot-proof: higher education in the age of artificial intelligence*. MIT press, 2017.

[135] Alberto Acerbi. *Cultural evolution in the digital age*. Oxford University Press, 2019.

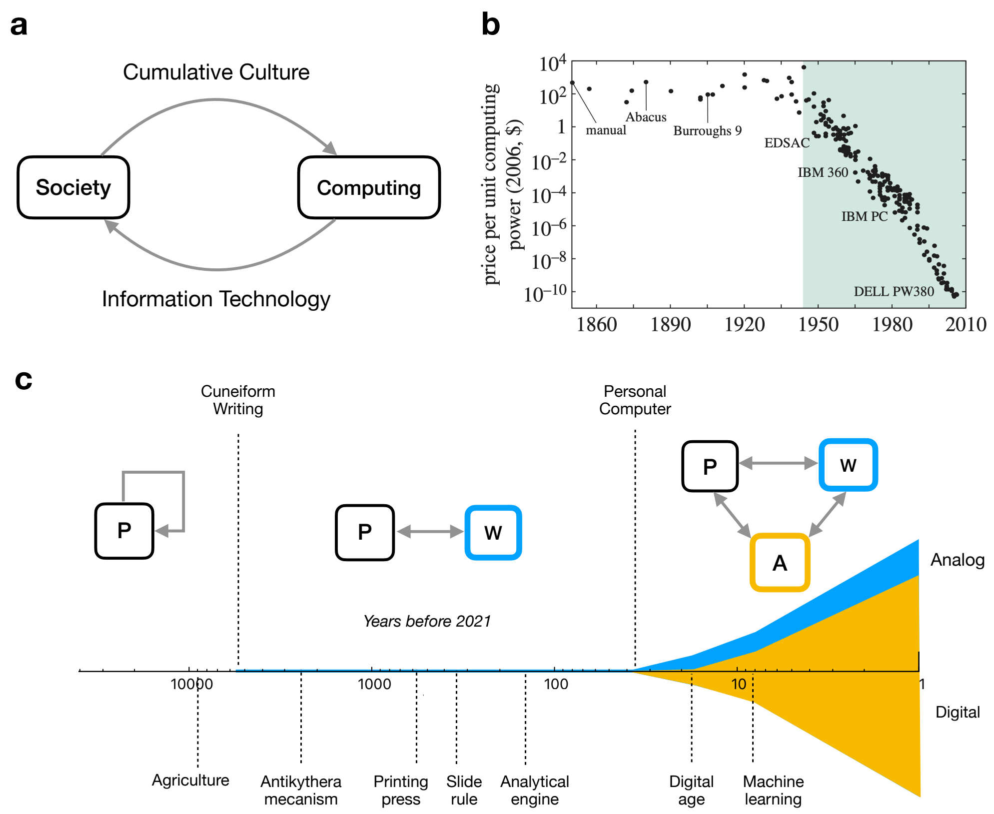

Figure 1: Acceleration of cultural change. (a) Co-evolution of society and computing driven by cumulative culture and information technologies. (b) Evolutionary pattern of computational costs across the years. A discontinuity exists in terms of scaling after the development of the electronic computer (around 1947). (c) Hyperbolic growth of knowledge caused by synergistic interactions between humans (P), material culture (W) and algorithms (A) (see text).

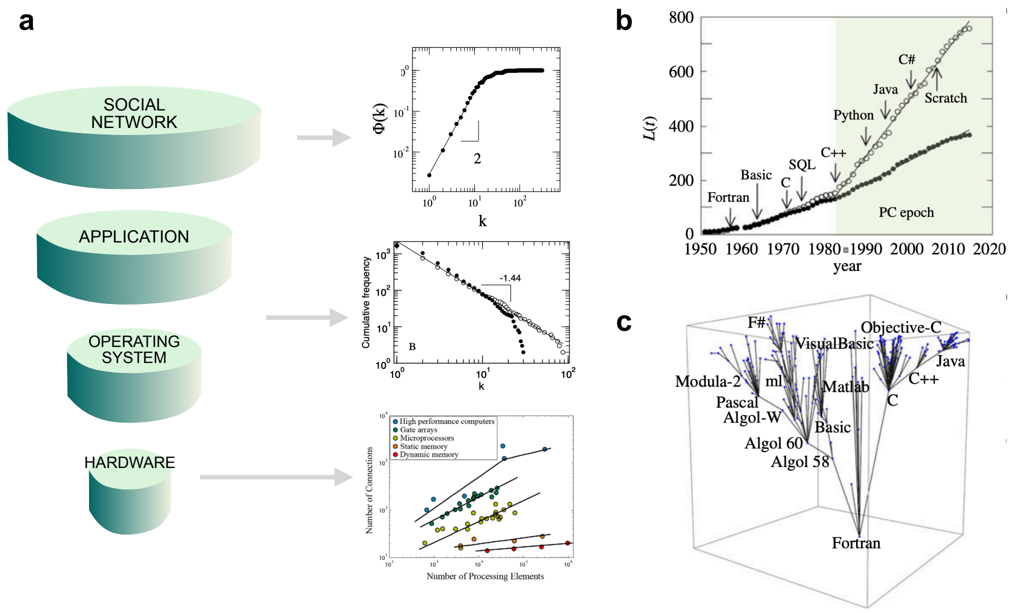

Figure 2: Network properties signal interactions between different mechanisms of cumulative cultural evolution and the major transitions in computing. (a) The multiscale organization of computing systems consists of (on the left, from bottom to top) hardware, software (which can be further decomposed in operating system and application), and social network. Each layer is associated with structural patterns (on the right, from bottom to top): rentian scaling, power-law connectivity distribution $P(k)$ and rich-club coefficient $\Phi(k)$ (see text). (b) Discontinuities in evolutionary trends reflect major changes in the cultural dynamics of computational systems, for example in the rates of recombination amongst programming languages. (c) The tree of technological influence also recapitulates the underlying process of punctuated dynamics, with sudden bursts of innovation.