Karp R M. Understanding science through the computational lens. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(4): 569–577 July 2011. DOI 10.1007/s11390-011-1157-0

Understanding Science Through the Computational Lens

Richard M. Karp

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1776, U.S.A.

E-mail: karp@cs.berkeley.edu

Received December 18, 2008; revised April 29, 2011.

Abstract This article explores the changing nature of the interaction between computer science and the natural and social sciences. After briefly tracing the history of scientific computation, the article presents the concept of *computational lens*, a metaphor for a new relationship that is emerging between the world of computation and the world of the sciences. Our main thesis is that, in many scientific fields, the processes being studied can be viewed as *computational* in nature, in the sense that the processes perform dynamic transformations on information represented as digital data. Viewing natural or engineered systems through the lens of their computational requirements or capabilities provides new insights and ways of thinking. A number of examples are discussed in support of this thesis. The examples are from various fields, including quantum computing, statistical physics, the World Wide Web and the Internet, mathematics, and computational molecular biology.

Keywords computational process, computer science and other sciences, computational lens

1 A Brief History of Scientific Computing

This article focuses on the changing nature of the interaction between computer science and the natural & social sciences. We begin by tracing a little of scientific computation history.

Classically, scientific computation has been associated with numerical analysis and focused on the solution of ordinary and partial differential equations and systems of algebraic equations, which arise in the modeling of physical problems.

But over time, the influence and methods of computation in the service of sciences are expanding. We now speak of computational science more broadly as dealing not only with the solution of systems of equations describing physical phenomena, but also with the simulation of complex computational models, along with various methods for the visualization of the results, including even virtual reality.

Another extension of the role of computation in sciences comes under the heading of *e-science*. This is an area characterized by employing use of data management tools to organize and analyze large masses of empirically obtained scientific data. E-science is a computationally intensive science, typically carried out in a highly distributed network and using very large datasets. Examples include the Sloan Sky Survey[1], and other data management systems for storing and accessing large bodies of climate data, oceanographic data, seismic data, and so forth.

2 Computational Lens

The above are some of the recognized interconnections between the world of computation and the world of science. This article focuses on a different relationship between the two fields, which we refer to as the *computational lens.*

In many scientific fields, the natural processes being studied are certainly based on physical transformations and transformations of energy, which is the way they have traditionally been viewed. But they can also be viewed as *computational* in nature, in the sense that natural processes perform dynamic transformations on information represented as digital or numerical data.

Through the computational lens, we can view natural or engineered systems arising in physical sciences, in engineering, or even in social sciences, in terms of their computational requirements and the way they transform information. This view allows us to apply the concepts of computer science to giving new insights and new ways of thinking.

Here are some examples of processes, which are physical, on the one hand, but can also be viewed as

Regular Paper

This work is supported in part by the National Science Foundation of USA for SGER under Grant No. CCF-0652536 "Planning for a Cross-Cutting Initiative in Computational Discovery" and Einstein Professorship of Chinese Academy of Sciences.

transforming information, and can be described by algorithms: regulation of protein production, metabolism and embryonic development, phase transitions of physical systems, mechanisms of learning, molecular self-assembly, strategic behavior of companies, and evolution of Web-based social networks.

We can think of the evolution of physical systems, such as collections of interacting magnetic spins, as undergoing computational transformations as they approach equilibrium.

Networks of proteins that regulate the activities of living cells can be viewed in terms of how they process information. We can think of behaviors as embryonic development as a computational process, in which individual cells of an emerging embryo determine their own specialized functions by processing information impinging on them from their environment.

Although we understand little of the mechanisms of learning in human or animal brains, we can at least dimly see that these mechanisms are inherently algorithmic.

In molecular self-assembly, a collection of molecules gradually interact, in a predictable way, to form complex structures. This can be described as an algorithmic process.

Not only in the physical world but also in the world of society and economics, we can think of behaviors as computational processes, such as the strategic behavior of companies and the evolution of markets, the determination of prices and organization of transactions. Similarly, the evolution of social networks on the Web can also be viewed in computational terms.

In the rest of the article, we discuss in more details a number of examples from different areas, and examine how they can be viewed as computational in nature.

3 Computational Lens at Berkeley

The theoretical computer science group at the University of California at Berkeley has long been working in the traditional areas such as computational complexity, cryptography, and algorithm design. But over the last several years, we have put a great emphasis on connections between the theory of computation and different scientific fields[2]. Professor Christos Papadimitriou[3-4] tries to understand the interactions among competing computationally limited agents in the World Wide Web and the Internet in terms of game theory, and studies the computational complexity of economic decision made from a game-theoretic point of view[3]. Professor Umesh Vazirani[5] seeks to understand the connection between models of quantum computing and the fundamental principles of quantum physics. Professors Sinclair and Mossel[6] work on connections between statistical physics and related statistical problems in computer science. My own work[7-8] along with that of many colleagues has been related to computational molecular biology. We try to understand how cells compute, in order to determine their behaviors, as a function of their environment.

3.1 Quantum Complexity Theory

Quantum complexity theory refers to the attempt to build quantum computers, and to construct models of computation that are faithful to the principles of quantum physics.

One might say that quantum complexity theory is "*the study of what we cannot do with computers we don't have*"[9]. It is a study of what we cannot do, in the sense that it tries to determine the ultimate limitations of computers, in this case, computers based on quantum physical principles. Since quantum computers have not yet been built up physically, we work with mathematical models of how a quantum computer might be realized in a manner consistent with the laws of physics.

The possible advantages of a quantum computer over a classical computing device can be understood by looking at the basic unit of information at the quantum level.

Quantum computation theory arises from the belief that at the most basic level, when we are dealing with small entities in physics such as atoms and atomic spins, the laws of quantum mechanics have to be respected. This belief leads to a different notion of computation. The basic unit of information is not the *bit*, which takes on only the values of 0 and 1, but a more complicated entity called *qubit*. A qubit is described by a pair of complex numbers associated with the possible states 0 and 1, which determine the probability when we actually observe the qubit that we will see the value 0 or the value 1.

More formally, a qubit can be described as $\alpha|0\rangle+\beta|1\rangle$ where α and β are complex amplitudes such that $|\alpha|^2 + |\beta|^2 = 1$. Observation of the qubit yields $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$, and the qubit collapses to whichever outcome we see. A quantum state of n qubits takes 2^n complex numbers to describe. Quantum logic gates perform unitary linear transformations on these amplitudes. Quantum computing tries to exploit this exponentiality for efficient computation.

Albert Einstein never accepted this probabilistic interpretation of subatomic computation; hence his famous dictum "*God does not play dice with the universe.*"[10]

A qubit contains much more information than an ordinary two-state bit of 0 or 1, because it has complex amplitudes associated with the basic states 0 and 1. However, we are limited in our ability to observe these complex amplitudes, since the observation affects the state. When we observe a qubit, all we will see is either the pure state 0 or the pure state 1, rather than a mixture of amplitudes. The goal of quantum computing is to exploit the massive amount of information stored in qubits to achieve efficient computation, in spite of the limits on observability.

It was an open question for some time whether the hypothetical quantum computer, based on qubits as the units of information, could perform computations more efficiently than the classical computer. This question was partially answered when computer scientist Peter Shor showed that integers can be factored into their prime factors in polynomial time on a quantum computer[11]. This is a computational task that we do not believe can be performed in polynomial time on a classical computing device although we have not proved the impossibility of doing so. Shor's results gave us a strong reason to think that quantum computers could be more powerful than classical computers. Also, because cryptographic systems, which are the basis of electronic commerce, depend on the intractability of factoring large integers, it follows from Shor's results that if we could build up quantum computers, then the systems of secure electronic transactions that our community depends on, will not be safe or secure.

The attempt to build quantum computers is the most severe test yet of whether the standard model of quantum physics is correct. If it is, then there is no impediment in principle to realizing a quantum computer. As Vazirani has said[5]: "*Quantum computing is as much about testing quantum physics as it is about building powerful computers.*" In the attempt to build quantum computers, we may be able to prove or disprove the validity of the standard model of quantum physics.

3.2 Statistical Physics

Statistical physics has a lot in common with certain areas of computer science, because both fields study how the behaviors of large systems of interacting entities can emerge from local interactions. In the case of statistical physical systems, these entities might be water molecules or magnetic spins, and the properties might be the freezing of water or the magnetization of magnetic material. In computer science, the interacting entities might be the constraints that arise in a constraint-satisfaction problem such as the Boolean satisfiability problem, or they might be the behaviors of a large number of participants in a social or economic network enabled by the Web. In both cases, we have systems with very large numbers of entities which cannot be completely observed. So we try to model them stochastically. The mathematics of analyzing these stochastic models in physics is very similar to what we use in related problems in computer science. Probabilistic models capture the statistical behavior of large, complex, heterogeneous, and incompletely known systems.

In statistical physics, a fundamental concept is that phase transition, in which the behavior of a system of interacting particles changes radically when a certain variable passes through a threshold value, such as temperature or strength of an external magnetic field. We have sharp phase transitions such as freezing of water or magnetization of a metal rod. In computer science, we have similar cases where the behavior of a computational system changes radically at a particular value of a parameter. For example, consider the classical satisfiability problem of Boolean formulas. If we look at random Boolean formulas, there is a critical value below which the formula is almost certainly satisfiable and above which the formula is almost certainly not satisfiable. That parameter is the ratio between the number of constraints and the number of variables. If the ratio of the number of clauses over the number of variables in a Boolean formula is above this threshold, then the formula is almost certainly unsatisfiable, and below the threshold, almost certainly satisfiable. This is very analogous to the phase transition of a physical system. There are many areas where the same kind of mathematics applies, to both to statistical physics and the computational models: constraint satisfaction problems, belief propagation and error-correcting codes, Markov chain Monte Carlo, percolation and sensor networks.

A famous result in computer science is the randomized polynomial-time algorithm for computing the permanent of a non-negative matrix, due to Jerrum, Sinclair, and Vigoda[12]. This problem includes, as a special case, the problem of counting the number of perfect matchings in a bipartite graph. The randomized polynomial-time algorithm for solving this problem in computer science is based on a technique arising originally in physics, Markov chain Monte Carlo, a method invented for sampling the states of a system of interacting particles.

The best algorithm currently known for solving very large random Boolean satisfiability problems is a technique called survey propagation[13], which was invented by statistical physicists, and can be thought of as a generalization of a "belief propagation" technique from computational learning theory.

So here we see very strong interplay between the two communities, statistical physics on the one hand, and computational learning theory and computational complexity theory on the other.

3.3 The World Wide Web and the Internet

Even though the Web and the Internet were built by humans, we cannot say that they were built according to a plan, because both the Web and the Internet simply grew under the influence and the decisions of many participants. We cannot think of them as designed by any particular designer, but rather as systems that just evolved through the independent activities of many individuals. Therefore, they have to be viewed as natural systems, and they have to be studied and understood empirically. This viewpoint on the Web and the Internet was stated in the following way by Christos Papadimitriou[3]: "*For the first time, we had to approach an artifact,* [*the Web or the Internet*], *with the same puzzlement, with the same uncertainty and understanding, with which the pioneers of other sciences had to approach the universe, the cell, the brain, and the market.*" In other words, we have to understand these artifacts by observation, rather than by looking at a prescribed design.

The Web and the Internet are computational systems, but they have other aspects as well. They are communication systems. They are social systems. They are economic systems. They support new modes of interaction between participating agents who communicate, collaborate, and undergo economic transactions and social interactions, on a large scale across the Web. The study of these systems gives rise to novel algorithmic problems: ranking the answers to a query submitted to a search engine, assessing the reputations of participants in economic transactions, recommendation systems, the design of auctions conducted over the Web, and the strategy of placing advertisements on web pages. All of these are algorithmic problems that have no counterparts in classical computation but arise because of the existence of the Web and the Internet.

This new medium for computation, communication, and social/economic interactions gives rise to many challenges, even for the social sciences. For example, because it is possible to record so much information about the evolution of a social network on the Web, the Web becomes a laboratory for sociologists and other social scientists. They try to answer questions about networks of interactions that link organizations and form communities of individuals. In a social network on the Web, how do ideas, opinions, innovations and technologies spread under the influence of communication between pairs of individuals? If we look at a community of interlinked web pages, how do we identify coherent "sub-communities"? If we look at all of web pages that have to do with the theory of evolution, how do we automatically identify sub-communities such as those who believe in evolution and those who believe instead in intelligent design? How can we understand the phenomena of "six-degrees of separation" in large networks that have evolved in the Web? In designing systems, such as peer-to-peer systems, how do we design them so that this six-degrees of separation phenomenon occurs, and how can we exploit it for the rapid accessing of information?

There are also challenges for economics, because in the world of e-commerce, it is necessary to invent new economic mechanisms. By an economic mechanism we mean an algorithm whose inputs come from economic agents with private data and selfish interests. Examples would be the individuals participating in an on-line auction, or bidding for the placement of Google ads. Economic mechanism design is concerned with ways of giving the participants incentives that will lead them to respond in the ways that the designer of the mechanism prefers. For instance, in designing an auction, one would like to induce the participants to behave in such a way that profit is maximized, or, in some other situations, one may want to maximize social welfare.

Economic mechanism design for various economic transactions over the Web has become an active area related to the field of computational game theory. Classical game theory is concerned with the study of rational behavior in situations of conflict. An example would be the arms race of the post war era where the Soviet Union and the United States were adversaries. Classical game theory and classical economics tended to assume perfect rationality on the part of participants. But in the newly developing area of computational game theory, we also take into account the computational complexity of strategic behaviors, and the limitations which prevent participants from behaving with perfect rationality because of their limits on their ability to compute.

One of the foundations of classical game theory is the concept of *Nash equilibrium* in a system, where the payoffs to any individual depend on the behavior of all of them, and where each individual has a choice of pursuing different strategies. The Nash equilibrium is an assignment of randomized strategies to all the players with the property that no single player will be motivated to change, to deviate from that strategy, as long as the other players do not deviate. There is a classical result that every game has a Nash equilibrium. A Nash equilibrium can be thought of as a prescription for the correct behavior of individual players.

But recently it was proven that computing a Nash equilibrium is a very difficult problem. Specifically, it is as hard as the problem of computing a Brouwer fixed point of a mathematical function[14]. This casts doubt on whether the Nash equilibrium is the appropriate model for the behavior of participants in complex situations of conflict, because it may be difficult to compute that behavior. As Kamal Jain put it[15]: "*If my laptop cannot compute it, neither can the market.*" The classical theory of market equilibrium is influenced by these computational limitations, and the formulation may have to be changed.

3.4 Pure Mathematics

Another area where there are strong influences between the theory of computing and a scientific field is pure mathematics. Mathematicians are increasingly aware of fundamental concepts from computer science, such as the P vs. NP question, NP completeness, randomized algorithms, derandomization (the process of converting a randomized algorithm to an equivalent deterministic one), public key encryption, and the complexity of factoring. Such issues originated in theoretical computer science, but are now commonly known among mathematicians in all areas of mathematics, and are influencing the questions that mathematicians ask. On the other hand, many aspects of modern mathematics have found a role in theoretical computer science. Metric space embeddings, random walks, Fourier analysis and other mathematical concepts, have become tools for theoretical computer science. So we see a very close interplay between computer science and pure mathematics.

4 Computational Processes in Biology

The rest of the article concentrates on the field in which it is the most productive to view Nature through the lens of computation.

4.1 Biology Computes at Many Levels

A few examples of computational processes are listed below, at different levels in biological systems. They are also physical and chemical processes. But in many aspects, they can be understood in terms of computation they inherently are performing.

- Learning in animal brains.
- The response of the immune system to an invading microbe, where the system senses the invasion and creates appropriate antibodies to neutralize the invading bacterium or virus.
- Specialization of cells during embryonic development, in which each cell within the body of the organism learns its appropriate role, whether it is going to be part of a wing or a leg or some other part of the body.
- The collective behaviors of animal communities, e.g., how birds organize themselves in a V-shape pattern as they fly, or how ant colonies or bee hives organize themselves to have specialized behaviors.
- Synthetic biology, where we try to design sensor-actuator control systems for regulation of biological processes by building control circuitry into the DNA of an organism to induce it to perform functions that were not originally intended by Nature, such as coercing bacteria to manufacture a drug that is used to treat malaria.

4.2 Goals of Computational Molecular Biology

Biology is undergoing a revolution. Advances in computation, experimental instrumentation and data gathering enable us to give a quantitative, algorithmic characterization of the processes which take place in biological systems. This opportunity to advance the understanding of molecular processes of life will also affect the way we diagnose and treat diseases. We may be able to understand in much more detail how the particular genomic makeup of an individual affects the individual's response to medicines, and therefore enables us to treat the diseases based on models of a particular individual's genome.

Biology is becoming a multidisciplinary field, involving not only the biological sciences, but also the physical, engineering and mathematical sciences, as well as the study of algorithms in computer science.

In particular, the emerging field of computational molecular biology has identified several goals of research:

- sequencing and comparing the genomes of many organisms;
- identifying the genes and determining the functions of the proteins they encode;
- understanding how genes, proteins and other molecules work together in an organized fashion to control the processes of the cell;
- tracing the evolutionary history and evolutionary relationships among existing species;
- understanding the structure and function of proteins;
- identifying the associations between genetic mutations and diseases. This area has become quite important recently. We now have the ability to analyze large databases of individual variations. We can take populations of individuals with and without a given disease, and look at how their genomes differ, and how differences of those genomes are correlated with the

tendency to have a particular disease.

4.3 Regulation in Molecular Biology

We need to recall some basic facts about molecular biology. In a eukaryotic cell, a part of the cell is walled off by membrane, and forms the nucleus of the cell, and the other parts of the cell are called cytoplasm. The DNA that constitutes our genome resides in the nucleus.

The fundamental dogma of molecular biology is that sequences within the DNA get transcribed into an intermediate form, called messenger RNA (mRNA), and the mRNA then gets transported to molecular machines called ribosomes outside the nucleus, and there a process of translation takes place, to translate the RNA sequences into protein.

The process of transcription from DNA to mRNA produces a one-to-one copy of the DNA. DNA is a sequence of four-nucleotides `A`, `C`, `T` and `G`. RNA is also a sequence of four nucleotides corresponding one-to-one to the DNA nucleotides. Transcription is just the direct writing of DNA in the language of the RNA nucleotides.

The process of translation from RNA to protein is more complicated. A protein molecule is composed of units called amino acids, of which there are 20 types. The translation from RNA to protein has been found to take place according to a universal code, which maps triplets of RNA molecules to the 20 different amino acids. This code is essentially the same in all living organisms.

In each cell there are thousands of different kinds of proteins that do most of the work of keeping the cell alive and functioning properly. But most proteins are transient molecules that last for only a couple of minutes to a few hours, and therefore have to be replenished when needed.

To understand how cells work we have to understand pathways and networks of interacting bio-molecules, DNA, RNA, and protein. The challenge of understanding these pathways was well stated by the biologist Garrett Odell[9]: "*We can approach understanding how the whole genome works by breaking it down into groups of genes that interact strongly with each other. Once researchers identify and understand these network modules, the next step will be to figure out the interactions within networks of networks, and so on until we eventually understand how the whole genome works, many years from now.*"

4.4 Regulation of Gene Expression

Animals are highly complex precisely regulated spatial and temporal arrays of differential gene expression. Gene expression refers to the process of manufacturing proteins associated with particular genes. Differential gene expression means that different cells and different environments express different genes and in different amounts.

This process of gene expression is regulated by complex networks of interactions among proteins, DNA and RNA. The challenge is to obtain an algorithmic description of how these pathways operate.

We discuss five levels below.

- At the *genome level*, the DNA contains the genes which spell the names of the proteins. The DNA is not itself active, but it can be thought of as a passive storage repository, where the genes are stored, implicitly describing the proteins that can be created.

- At the *transcription* level, the transcription of genes to mRNA is regulated by the binding of certain proteins called *transcription factors* to DNA in the control regions of genes. We need to describe the combinatorial processes by which the abundance of these transcription factors controls the transcription of the genes to RNA.

- At the *translation level*, the translation of mRNA into functioning proteins is regulated by complex networks of protein-protein and protein-RNA interactions, and by post-translational modifications of proteins.

- Another level is the actual *metabolic processes* taking place within the cells. Regulation of metabolic processes involves complex networks of chemical reactions catalyzed by proteins called enzymes.

- All of these processes together lead to global phenotypes, and global behaviors, such as diseases, which are regulated by the *interaction* of many of these chemical processes.

So we have the genome, the regulation of transcription, the regulation of translation, the influence of proteins controlling chemical processes in the cells, and interactions of all these processes, which collectively influence behaviors.

Several types of tools are available for analyzing these processes. On the experimental side, we have *large scale experimental measurement* tools for measuring protein-DNA interactions, the bindings of proteins to each other, and the levels of mRNA production under various, perturbed conditions in a cell. On the computational side, we have *DNA sequence analysis* to identify the genes, to find the regulatory regions associated with these genes, to identify the transcription factors and the places within the genome where the transcription factors bind. *Phylogenetic analysis* identifies regulatory structures conserved across species. We try to compare the regulatory structures in different species, because one of the principles of biology is that once Nature solves a problem in one species, it is likely to use similar mechanisms in other related species. Yet another type of tool is the *classification* of proteins according to their structures and functions.

These tools help us understand the fundamental types of regulation. One task is the analysis of *protein-DNA interactions*, with the goal of breaking the cis-regulatory code, i.e., understanding how transcription factors, binding to DNA near the start site of the transcription of a gene, influence the transcription. This was described by the renowned biologist Eric Davidson as follows[17]: "*Regulatory interactions mandated by circuitry encoded in the genome determine whether each gene is expressed in each cell, throughout developmental space and time, and, if so, at what amplitude.*"

A second task is the analysis of *protein-protein interactions* for identification of molecular machines and signal transduction cascades. There are large databases now available based on measurements of which pairs of proteins bind together. Given this information, we can look for highly interacting groups of proteins, which have the property that these interactions occur in several different organisms, leading to evidence that these interacting sets of proteins are performing basic functions of the cell.

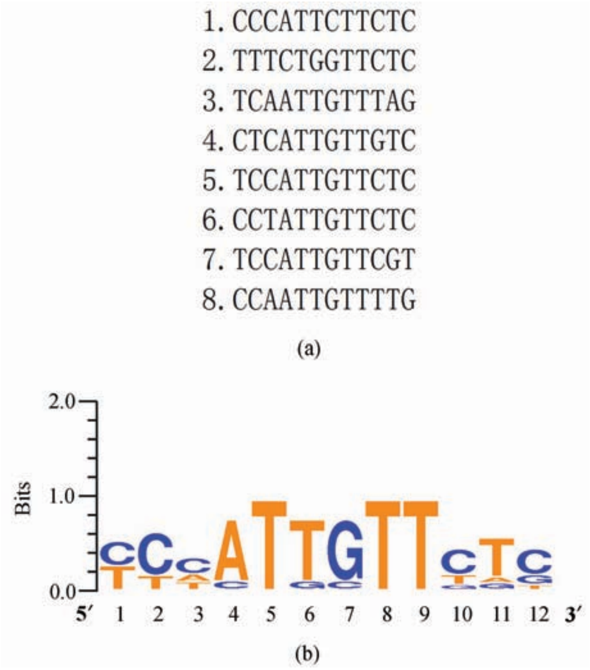
Let us first look at analysis of protein-DNA interactions. Recall that transcription is regulated by proteins called transcription factors that bind to DNA near the start site of the transcription of a gene, and our goals are to understand this as an algorithmic process. We need to identify the transcription factors, to characterize the sites they bind to in the genome, and to determine how the transcription factors act in combination to enhance or limit transcription. This information is referred to as the *cis-regulatory code.*

In complex organisms such as man, the gene does not consist of a single DNA sequence, but is composed of pieces called *exons*, separated by intervening *introns*. In the process of transcription, the whole sequence becomes transcribed into mRNA, the introns are then removed, and the exons are spliced together to determine the eventual transcript that determines the protein production. Several binding sites appear before the start site of transcription, where different transcription factors bind to the DNA. `CCAAT` and `TATA` are typical DNA sequences that these transcription factors recognize.

This leads to two challenges in order to understand this process correctly. One is to understand the sequences that different transcription factors recognize; and the other is to understand the combinatorial patterns that many different transcription factors work together to determine the level of transcription of a gene. The problem is complicated by the fact that the sequences that these transcription factors recognize are not only exact sequences, but there is some possibility of variations, which does not interfere with the recognition of the sequences.

Our goal is to identify *motifs*, i.e., the short sequence patterns recognized by a transcription factor. These motifs occur repeatedly in the genome, but with considerable stochastic variation. Some positions are highly conserved, others exhibit great variation. Certain combinations of motifs occur repeatedly in clusters. Thus having identified motifs, we would like to understand how combinations of these motifs affect transcription.

Fig.1 shows an example motif[18]. A motif can occur on either strand of the DNA, so it can be read in either one direction or the reverse direction, because DNA is double stranded. We see that all these sequences have general similarity, although the motif's occurrences are not identical.



1. CCCATTCTTCTC
2. TTTCTGGTTCTC
3. TCAATTGTTTAG
4. CTCATTGTTGTC
5. TCCATTGTTCTC
6. CCTATTGTTCTC
7. TCCATTGTTCGT
8. CCAATTGTTTTG

(a)

(b)

Fig.1. A motif example. (a) Eight motif copies. (b) Logo of the eight motifs.

A motif can be thought of as a kind of probabilistic word, which is sometimes depicted by a diagram, such as the one shown in Fig.1(b), called *logo.* The large symbol `T` on the fifth position indicates that this position is *conserved*, that a copy of the motif has to have the nucleotide `T` in this position, as shown in Fig.1(a). But in some of the other positions, there is statistical variation. The motif can have either an `A` or a `C` in the fourth position, with `A` being more frequent as indicated by the bigger size of the symbol.

A regulatory module is a set of mutually cooperating transcription factors that can bind to the control regions of genes to enhance or inhibit transcription. Given a database of transcription factors and their binding motifs, our task is to identify such modules by searching for sets of transcription factors whose binding sites tend to co-occur in control regions.

Regarding algorithmic description of a cis-regulatory network, Eric Davidson put it as follows[17]: "*Portions of the endo16 cis-regulatory system of Strongylocentrotus are to date the most extensively explored of any, with respect to the functional meaning of each interaction that takes place within them. What emerges is almost astounding: a network of logic interactions programmed into the DNA sequence that amounts essentially to a hardwired biological computational device.*"

Another area of active research is protein networks, or analysis of protein-protein interactions. Databases of protein-protein interactions are now available for several species. Searches through protein sequence databases reveal similarities between proteins in different species. Furthermore, many protein interaction networks, including protein complexes and signaling pathways, are conserved: they have evolved over evolutionary time and occur, in modified forms, in many organisms. Collections of proteins bind together to form molecular machines that operate across several different organisms.

Our goal is to identify conserved protein complexes and signaling pathways, using databases of protein-protein interactions in several species, in conjunction with data about protein sequence, structure, function and expression. Discovery of conserved pathways and complexes allows transferring of functional annotation and prediction of interactions from one species to another.

Research results have been obtained indicating putative complexes conserved in yeast, worm, and fly. 173 complexes have been identified in the three organisms. These conserved complexes are associated with the following functions: DNA, RNA and phosphorus metabolism, intracellular transport, regulation of transcription, protein folding, synthesis and degradation, homeostasis, cell proliferation, development and growth, and RNA localization.

5 Conclusions

In many different fields of sciences, both physical sciences and social sciences, the basic underlying processes can be thought of as *computational*, and can be analyzed through the lens of computer science.

The power of this computational perspective is multi-faceted: it exposes the computational nature of natural processes and provides a language for their description. It brings to bear fundamental algorithmic concepts, such as adversarial and probabilistic models, asymptotic analysis, intractability, computational learning theory, threshold behavior, fault tolerance. It alters the worldviews of many scientific fields.

This algorithmic worldview is changing the sciences, including mathematical science, natural science, life science, and even social science. Computer science is placing itself at the center of scientific discourse and exchange of ideas.

Acknowledgements This article is based on the Einstein Professorship Lecture of Chinese Academy of Sciences, given by the author in Beijing on June 18, 2008. The author would like to thank Jane Yang, Wei-Kun Zhou, Bo Yan, and Dong-Bo Bu of Chinese Academy of Sciences for transcribing his talk into text form.

References

[1] Sloan Digital Sky Survey. http://www.sdss.org/.

[2] The Group of Theory at Berkeley Website. http://theory.cs.berkeley.edu/.

[3] Papadimitriou C H. The algorithmic lens: How the computational perspective is transforming the sciences. In *2007 Federated Computing Research Conference*, Speech, San Diego, USA, Jun. 8-16, 2007.

[4] Papadimitriou C H. Algorithms, Games, and the Internet. In *Proc. STOC/ICALP 2001*, Heraklion, Greece, July 6-8, 2001, pp.749-753.

[5] van Dam W, Mosca M, Vazirani U. How powerful is adiabatic quantum computation. In *Proc. Symposium on the Foundation of Computer Science*, Las Vegas, USA, Oct. 14-17, 2001, p.279.

[6] Mossel E, Peres Y, Sinclair A. Shuffling by semi-random transpositions. Mathematics arXiv math.PR/0404438, April 2004, Conference version appeared in *Proc. IEEE FOCS 2004*, Rome, Italy, Oct. 17-19, pp.572-581.

[7] Xing E P, Karp R M. MotifPrototyper: A Bayesian profile model for motif families. *Proc. Nat. Acad. Sci.*, USA, Jul. 20, 2004, 101(29): 10523-10528.

[8] Ben-Dor A, Chor B, Karp R, Yakhini Z. Discovering local structure in gene expression data: The order preserving submatrix problem. *Journal of Computational Biology*, 2003, 10(3/4): 373-384.

[9] Aaronson S. The Limits of Quantum Computers. http://www.scottaaronson.com/talks/sipbtalk.ppt.

[10] Albert Einstein. Letter to Max Born (4 December 1926). The Born-Einstein Letters, translated by Irene Born, New York: Walker and Company, 1971, ISBN 0-8027-0326-7. This quote is commonly paraphrased "God does not play dice" or "God does not play dice with the universe", and other slight variants.

[11] Shor P. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proc. the 35th Annual Symposium on Foundations of Computer Science*, Los Alamitos, USA, Nov. 20-22, 1994, pp.124-134.

[12] Jerrum M, Sinclair A, Vigoda E. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *Journal of the ACM*, 2004, 51(4): 671-697.

[13] Braunstein A, Mézard M, Zecchina R. Survey propagation: An algorithm for satisfiability. *Random structures &*

Algorithms, 2005, 27(2): 201-226.

[14] Etessami K, Yannakakis M. On the complexity of Nash equilibria and other fixed points. In *Proc. the 48th IEEE Symp. Foundations of Computer Science* (*FOCS*), Providence, USA, Oct. 20-23, 2007, pp.113-123.

[15] Nisan N, Roughgarden T, Tardos E, Vazirani V V (eds.). Algorithmic Game Theory. Cambridge University Press, 2007.

[16] Cipra B A. Some assembly required. SIAM News. Dec. 1, 2003, 36(10).

[17] Davidson E H. Genomic Regulatory Systems: Development and Evolution. Academic Press, 2001.

[18] Patrik D'haeseleer. What are DNA sequence motifs? *Nature Biotechnology*, 2006, 24: 423-425.

Richard M. Karp received his Ph.D. degree from Harvard University in 1959. He is currently a professor at the University of California, Berkeley, and a research scientist at the International Computer Science Institute in Berkeley. The unifying theme in Karp's work has been the study of combinatorial algorithms. His current activities center around algorithmic methods in genomics and computer networking. His honors and awards include U.S. National Medal of Science, Turing Award, Fulkerson Prize, Harvey Prize (Technion), Centennial Medal (Harvard), Lanchester Prize, Von Neumann Theory Prize, Von Neumann Lectureship, Distinguished Teaching Award (Berkeley), Faculty Research Lecturer (Berkeley), Miller Research Professor (Berkeley), Babbage Prize and eight honorary degrees. He is a member of the U.S. National Academies of Sciences and Engineering, the American Philosophical Society, the French Academy of Sciences, and a Fellow of the American Academy of Arts and Sciences, the American Association for the Advancement of Science, the Association for Computing Machinery and the Institute for Operations Research and Management Science.