The Computational Worldview and the Sciences: a Report on Two Workshops

Sanjeev Arora[*] *Avrim Blum*[†] *Leonard J. Schulman*[‡]
Alistair Sinclair[§] *Vijay V. Vazirani*[¶]

October 12, 2007

[*]Computer Science Dept, 35 Olden St, Princeton NJ 08540, arora@cs.princeton.edu
[†]Department of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213-3891. avrim@cs.cmu.edu
[‡]Caltech, MC256-80, Pasadena CA 91125, schulman@caltech.edu
[§]Computer Science Division, University of California, Berkeley, CA 94720-1776. sinclair@cs.berkeley.edu
[¶]College of Computing, Georgia Tech, Atlanta, GA 30332-0765, vazirani@cc.gatech.edu

Summary

In many natural sciences and fields of engineering the processes being studied are computational in nature: for example, protein production in living cells, neuronal processes in the brain, or activities of economic agents as they incorporating current prices and market behavior into their strategies. We believe that viewing natural or engineered systems through the lens of their computational requirements or capabilities, made rigorous through the theory of algorithms and computational complexity, has the potential to provide important new insights into these systems. This impact on scientific methodology is distinct from and complementary to the impact computers have had and will continue to have through optimization and scientific computing.

Thus the "algorithmic way of thinking" could play the role of a key enabling science of the 21st century, similar to and entwined with the role played by mathematics. For instance, within the past quarter-century, viewing quantum mechanics from a computational perspective gave rise to quantum computing, and viewing genomic sequencing as an algorithmic process rather than a wet lab process led to the fast sequencing of the human genome.

With the aid of funding from the National Science Foundation[1] we organized two workshops at Princeton University and Caltech whose stated goal was "*to identify and pursue novel insights that may be obtained by applying a computational worldview to the Natural, Social and Mathematical Sciences.*" Speakers from a dozen disciplines were invited "*to not only identify areas of scientific computation where new algorithms are needed, but also instances where computational concepts play a role in understanding the underlying phenomena.*

Talks from different disciplines often turned out to have overlapping themes, and it seems that Theoretical Computer Science (TCS) insights into efficiency, asymptotic analysis, universality, learning, fault tolerance, algorithmic and network phenomena, threshold behavior, etc., can provide crucial new perspectives in many of those open problems. This kind of work could be an important component of the upcoming CDI initiative at NSF.

The major open problems can be roughly partitioned into the following three subcategories. (Of course, some problems such as "understanding the brain" span two or even all three categories.)

Need for new models. In emerging areas such as nanotechnology, self-assembly, understanding the living cell's regulatory processes, or understanding strategic interactions in a variety of settings, it would be a good and important first step to clearly and succinctly model the process, and to understand basic properties of this model. Usually this model would be phrased in an algorithmic language and have associated precise complexity measures (which are sometimes omitted in traditional modeling).

Need for new modes of analysis, especially of global properties. In many areas, though we have a fairly clear idea of the local properties of a system or a process, we lack a good understanding of the global properties. For example, many physical systems (self-assembling nanostructures, flocks of birds or sensors, matter undergoing phase transitions, neurons in the brain, to name a few) can be viewed as networks of interacting agents. Apart from computer simulations, we currently for the most part lack any other way to understand such interactions qualitatively or quantitatively. New modes of analysis would represent big progress in understanding these systems. Usually these systems consist of discrete elements or agents, so classical continuous modeling (using, say, a PDE) is perhaps inappropriate. Often the agents' interactions involve strategic behavior, and

[1]NSF SGER grant CCF-0652536, "Planning for a cross-cutting initiative in computational discovery."

then the setting is reminiscent of *game theory*, except that there are important differences from the classical setting—such as the fact that agents cannot be assumed to be *rational* or *computationally all-powerful*. Thus an *algorithmic* theory of games—one that assumes computationally limited agents—is needed.

Need for new algorithms. The above two categories in themselves represent new directions for algorithm research. But there are also other problems—such as testing cosmological theories, or learning regulatory processes in the cell from available data—that call directly for design of new algorithms along traditional lines or at least new algorithmic thinking. It also seems that new modes of algorithmic thinking developed in the past decade or so—streaming algorithms, metric space embeddings, approximations, etc.—may also be usefully applied to these new problems.

Sample open problems

Is a crowd of quantum particles different from a few particles? This question comes from quantum mechanics and quantum computing. Standard phenomena in quantum mechanics such as QED, atomic and molecular structure, and Bell states and Bell inequalities involve only a small number of particles. Even Bose-Einstein condensates, though involving many particles, are effectively low-dimensional systems. On the other hand, the laws of quantum mechanics tell us that highly entangled quantum states will arise in various large many-body systems: for example Kitaev's honeycomb lattice and topological quantum computing. Is there a way to verify the theory of quantum computing in the context of these larger, more complex systems in a similar manner to the way that single particle systems have been verified to an exquisite level of accuracy? The realizability of quantum computers hinges on the answer. It may be argued that so does the standing of quantum mechanics as a predictive physical theory.

Understanding of non-equilibrium dynamics. This problem is articulated later in the section on statistical physics but a better understanding of this issue will probably have repercussions in many other areas. We know a lot about equilibrium properties of discrete probabilistic processes that are studied in statistical physics and other fields. It is an open problem to understanding non-equilibrium properties, including the approach to equilibrium and the behavior close to equilibrium states. This is relevant, for example, in designing and modeling molecular-scale engines and other computational devices. (See also the Nanotechnology section of this report.) This is an area where algorithm designers may be able to contribute novel ideas, since they may be able to look at the problem more broadly than scientists looking at it within the constraints imposed by physical systems.

Understanding of strategic interaction among resource-limited agents. Game theory has had spectacular success in economics and related fields. The chief idea underlying it is the notion of *equilibrium*, which usually assumes rational agents and often ignores issues of how the equilibrium is arrived at. Even so, there are good reasons to continue to use such equilibria for economic theory. However, it is increasingly clear that strategic interactions arise in all kinds of noneconomic domains, and that a rationally achieved equilibrium is not always an appropriate notion. A growing discipline called *algorithmic game theory* seeks to study strategic interactions among computationally bounded agents. These issues are also becoming increasingly critical to the field of *distributed control*.

Exploring the algorithmic power and limitations of new neural models. Experimental work in neuroscience has presented a much richer picture of neuronal processes than the one that was available a couple of decades ago, involving a rich mosaic of neurotransmitters and spiking behavior (see the Section on neuroscience). There is a need to explore the algorithmic possibilities of these new models of neuronal networks and to understand their computational limitations, which may help isolate tasks that they *can or cannot* do efficiently. This would help pinpoint inadequacies in our current understanding of neuronal processes.

Algorithms for learning/modeling regulatory processes in the cell. Understanding gene expression is an important goal for biology in the next decade. The regulatory processes involved in gene expression are usually expressed in a circuit-like manner. A first natural question is whether the current models can be enriched by taking a more computational or game-theoretic view. A second question is to design algorithms that can infer the regulatory process from experimental data (which would also suggest what kinds of experiments to perform in the first place).

New architectures and algorithms for nanoelectronic devices. Circuits and devices manufactured using molecular self-assembly are one of the end-goals of nanotechnology. For the foreseeable future, self-assembly has to deal with a significantly higher defect rate than etching and similar methods; this presumably has to be dealt with at the algorithmic level. Thus we need a theory of fault-tolerant assembly, as well as new fault-tolerant algorithms and architectures for these models.

Testing models in cosmology and other fields. Computational cosmology and astrophysics is a growing field that takes advantage of the ready availability of data from sky surveys to test cosmological theories. Clever algorithms have already played a role in this task, but it seems that recent algorithmic work on geometric embeddings and metric dimension may play an important role as well. Ideas from machine learning and property testing may also provide new concepts for what it means to "test" a theory. The viewpoint is equally applicable to other scientific fields in which theories need to account for massive, high-dimensional, noisy data.

Understanding pseudorandomness. This problem comes from mathematics. The question of how random-looking deterministic objects can be, and in what sense can we we call them pseudo-random, is fundamental in many areas of mathematics: statistics (theory of designs), number theory (distribution of primes, exponential sums), PDE's (regularity of wave propagation) and combinatorics (Ramsey theory, discrepancy theory). In theoretical computer science, a computational theory of pseudorandomness was developed, with applications to probabilistic algorithms, cryptography, computational complexity and weak random sources. Many connections between the above frameworks for pseudorandomness already exist, and some recent ones are particularly striking: use of the sum-product theorem to build randomness extractors, and from there to construct Ramsey graphs; use of Szemerédi's Regularity lemma in property testing and in finding long arithmetic progressions in primes, and the use of Gowers' uniformity in circuit lower bounds and pseudorandom generators.

Algorithmic modeling in the social sciences. A range of questions in the social sciences call for computational or algorithmic explanations. In particular, what are realistic models for

how the structure of a social network evolves, taking into account the collective behavior of the large population of interacting agents that comprises it? How can we use such models to explain and potentially predict the dynamics of social processes, such as the tendency for cascading "word-of-mouth" effects to propagate some ideas and innovations very widely through society, while other, similar ideas, never escape a small circle of people? And as an increasing number of the most prominent Internet information systems come to exhibit rich social structure (e.g. Facebook, Wikipedia, YouTube), how can we use algorithmic insights into social processes to guide the design of future systems, potentially strengthening on-line communities and ultimately leading to more productive discourse in social, economic, and political spheres?

1 Introduction

It is well recognized that computer science will be a key enabling science of the first half of the 21st century. This report draws attention to the fact that the algorithmic way of thinking, developed within computer science, also provides a new way of looking at problems in a host of other disciplines.

In many natural sciences and fields of engineering the processes being studied are computational in nature: for example, protein production in living cells, neuronal processes in the brain, or activities of economic agents as they incorporating current prices and market behavior into their strategies. We think that viewing natural or engineered systems through the lens of their computational requirements or capabilities, made rigorous through the theory of algorithms and computational complexity, provides important new insights into these systems. Theoretical Computer Science can play an important role in elucidating and exploiting these new insights, and furthermore, this has the potential to be an important new focus for computer science research and education in the coming decades.

With this viewpoint in mind, and with the help of an SGER Grant from NSF, we organized two workshops to which we invited a number of experts from a diverse collection of fields to expound problems in their fields which could benefit from an algorithmic way of thinking. A distinguishing feature of these workshops was their multidisciplinary nature. Speakers were asked to very deliberately focus on the computational lens and its use in solving problems from their field.

Of course, the TCS community has already been exploring, with moderate to considerable success, problems in several fields, including quantum computing, economics, statistical physics and biology. As detailed below, through these workshops, we were able to identify a number of promising areas which theoreticians have not explored yet or have only begun to explore.

We believe the successes achieved by the TCS community in this endeavour represent a mere beginning. As many as a dozen similar success stories could emerge within the next decade or two, as described below. These potentially represent an important new funding direction for CISE, and indeed for all of NSF. In addition to jump-starting research in a host of scientific disciplines, these research directions could also give a new focus to computer science research and training in the coming decades, and also have substantial economic impact within the US and abroad.

We note that the potential applicability of computer science to these disciplines derives from two core strengths: its experience with modeling and understanding computational phenomena, and its expertise in algorithmic thinking. We believe that TCS insights into efficiency, asymptotic analysis, universality, learning, fault tolerance, algorithmic and network phenomena, threshold

behavior, etc., will provide crucial new perspectives in many of these settings. This leads to an important educational component of our initiative: the education of a new (and larger) generation of graduate students within CS who integrate this understanding, and export it through collaborative research and education to the other sciences.

It is important to clarify one point for the rest of the report. "Algorithmic advances" should not be taken to mean simply a faster implementation of a known method – such advances will predictably occur, and TCS will of course play a role. It should not even be taken to mean "merely" finding faster algorithms for well defined computational problems – a central topic in disciplinary TCS research. We mean the phrase to include even more broadly the harder and scientifically more subtle process of developing new algorithmic frameworks and mathematical analyses that are applicable to scientific domains where information processing plays a role, but which are not yet well-addressed by the theory of computing. This is the truly fertile ground for new TCS insights.

Indeed, the main theme in this proposal is that the past successes and the problems identified below are not isolated examples: fundamental insights from the study of computation will increasingly enrich the rest of science. Indeed, algorithms will play an increasing role in efforts to develop a cyberinfrastructure for science and engineering. Furthermore, the list of problems below should also be seen as a call for more research in the core areas of TCS, which deserve major credit in the successes achieved so far in that they have provided a powerful arsenal of tools and techniques for attacking difficult, fundamental problems in other sciences.

In the Appendix of this report, we provide a list of all the talks given at the two workshops as well as a list of the attendees. We also give pointers to the workshop web-sites where videos and slides of the talks are available for download. Given the wealth of ideas expounded by the speakers, the latter form an important repository for the research community.

We conclude this introduction with two caveats. Due to the extremely broad scope of the material covered in this report, we have not attempted to be comprehensive in our treatment; rather, we have aimed to provide some representative examples to illustrate potential future areas for collaborative research. For the same reason, we have not attempted to provide bibliographic references; any meaningful list would of necessity be longer than the report itself. The interested reader may find many references in the talk slides and videos on the workshop web-sites.

2 Game Theory and Mathematical Economics

The link between game theory and computer science goes back all the way to the dawn of these two fields: in the work of von Neumann, who initiated game theory with his theorem on two-person zero-sum games and also introduced the notion of stored program computers. Some of the most prominent early researchers in game theory — Dantzig, Gale, Kuhn, Scarf, Shapley, and Tucker — also made fundamental contributions to the field of algorithm design. After this early synergy, however, these two fields grew in near isolation from each other for decades. Now, intriguingly, the two fields have come together again, motivated largely by the Internet and its associated scientific and commercial challenges.

Below we first outline the exciting issues being explored in the new area of algorithmic game theory (an area that was not represented at the workshops since its prominent open problems were already well known to us). We then summarize the problems discussed in talks by Andrew Postlewaite, Ehud Kalai and Andrei Broder at the Caltech workshop. The economics session also had talks by Colin Camerer and Antonio Rangel; since they had a substantial overlap with

neuroscience, these talks are discussed in Section 5.

2.1 Algorithmic Game Theory

The advent of the Internet has been the primary impetus for a renewed synergy between game theory and algorithms and the birth of algorithmic game theory (AGT) as a bona fide research area. The Internet can be thought of as a gigantic playground in which a myriad of diverse entities (players) are constantly involved in various types of interactions: sharing common resources, participating in the slew of markets that have arisen on the Internet, sharing digital goods such as files, music and videos, advertising their services and products, and so on. Many of these interactions are strategic and the outcome has enormous consequence for the individuals and companies involved. To ensure smooth functioning in these strategic situations, game theory is the natural field to turn to for finding good solutions.

Interactions on the Internet happen on very different scales from interactions between people, in terms of both the number of entities and the speeds involved. Additionally, the settings that need to be addressed, such as on-line auctions and markets, are becoming increasingly complex. On the positive side, the entities have available at their disposal moderate to massive computational power. For these reasons, a game theoretic solution can be considered viable only if it has good computational properties, such as running time and communication complexity. Although huge strides have been made in the last ten years on these issues, a lot of work still remains to be done. Below we identify four areas of ongoing research within AGT.

Equilibria and their Computation
One of the key solution concepts identified by game theory and economics was that of an equilibrium: a stable state in a strategic interaction. Nash and market equilibria are two primary examples. Although questions of existence and agents' behavior in equilibria have been studied extensively, these fields have traditionally neglected the question of how computationally easy — or difficult — it is for agents to arrive at equilibrium. This being a fundamental algorithmic question, it was naturally the first one to be attacked by researchers in TCS. Recent negative results obtained on computing Nash equilibria have opened up the central question of finding an alternative solution concept that is both economically meaningful and amenable to efficient algorithms.

Algorithmic Mechanism Design
The well developed area of mechanism design within game theory is undergoing a transformation into the area of algorithmic mechanism design at the hands of TCS researchers and their collaborators. This area attempts to design games at whose equilibria the designer's goals are achieved, independent of the agents' private valuations. Thus, for example, if the goal of the equilibrium it to ensure social welfare, then despite their selfish interests the players end up playing towards the common good of all! Such games are precisely what is needed for the smooth functioning of the Internet.

As an illustrative example, consider the notion of an auction, a classical concept that has undergone a transformation in light of the new situations in which it is being applied. Two prominent examples are auctions of *digital goods* and auctions of *keywords* in the search engine industry. Work on the former has yielded radically new ways of carrying out an auction. The latter is the subject of intense investigation and promises to have a huge impact, considering the fact that these auctions are the main source of revenues of the multi-billion dollar search engine industry. The phenomenal expansion of e-commerce is continually posing new challenges for algorithmic mechanism design.

Markets and their Computational Issues
The study of market equilibria has occupied center stage within mathematical economics for over a century and has resulted in such celebrated works as the Nobel prize winning Arrow-Debreu Theorem. The question of computability of these equilibria has been the subject of intense research within TCS in recent years. This study, conducted in an ancient arena from a new point of view, has already yielded new insights not only in AGT but also in the theory of algorithms. Impetus for this work is provided by the myriad of markets hosted on the Internet, both old and new (prominent among the latter being markets launched by eBay and Amazon, and Google's AdWords market), which already occupy a substantial fraction of the economy and are projected to grow in the future, and the massive computational power available for running them.

A somewhat less obvious question, but one with a potential for major impact, arises from the following observation. Paradoxically, whereas sophisticated principles are being used to optimally run the markets hosted on the Internet, its underlying connectivity market uses a primitive and highly inefficient system that results on the one hand in huge amounts of unused connectivity, while on the other hand there are end-systems that want to send traffic and are willing to pay, but have no mechanism for buying connectivity. It also results in needlessly long paths that use resources inefficiently. Recent proposals for establishing a transparent, free market for connectivity seem promising but still require substantial algorithmic and protocol development.

The Price of Anarchy
Finally, the notion of *price of anarchy*, developed within AGT, is expected to play a role in evaluating the efficiency of game theoretic solutions deployed on the Internet. The price of anarchy of a game-theoretic solution evaluates the inefficiency that results when agents arrive at an equilibrium, as compared to the situation in which they selflessly collaborate to minimize the total cost. Clearly, a solution concept with a large price of anarchy is not very useful, even if its equilibria can be found quickly.

2.2 The Talks

Kalai's talk on "Large Games" focused on his recent results showing the unusual robustness properties of games involving a large number of players. He showed that if such a game has a "semi-anonymous" payoff structure, then every equilibrium of the "single-shot" version of the game carries over to all reasonable "alterations" of the game. The potential applications to AGT are obvious, since games on the Internet do involve a large number of players.

There are several computationally relevant directions in which this research can be extended: Are equilibria in large games easier to compute, given their nice robustness properties? Can the theory of large games be applied to concrete computer science applications? Can similar strong results be derived even under weaker assumptions than the semi-anonymous payoff structure?

Postlewaite's talk discussed the issue that conventional modeling of how economic agents make decisions is hardly satisfactory for many real-life problems. Firstly, whereas the rationality paradigm requires the agents to make the best decisions given the information they actually or potentially know, computing the first best choice can be prohibitively hard. Postlewaite's focus is much deeper: the very understanding of the data — analysis, organization, and operation with — may also be hard. It is crucial for us to have a better understanding and modeling of how "realistic" agents can achieve this. In particular, Postlewaite showed that the computational problem underlying commonly used linear regression methods is NP-hard.

In view of this, the following questions arise: When do approximate solutions exist? Are there alternative specifications of the problem? If so, what do their cost functions look like? Are there applications to complicated (non-continuous) consumer choice problems, such as selecting a place to work, buying a car, etc.? How can one approach such a problem? What decision rules can one use?

In the future, computer scientists could profitably expand their focus from these issues to other areas of economic theory, such as business cycles, investment bubbles, exchange rate movements, etc., where aggregate behavior and network effects play an important role.

Andrei Broder's talk, "Technical challenges in web advertising" focused on the exciting new market for selling advertisements on the web, created by search engine companies such as Google and Yahoo!. This market is responsible for over 90% of Google's revenues and over 40% of Yahoo!'s revenues, hence making these search engines available for free to users. In addition, it has completely revolution advertising products and services to customers – making it highly targeted and opening up a venue for small businesses to pitch to their potential clients with very small expenditures. However, in typical "fat tail distribution" fashion, the number of these small advertisers is very large and is responsible for a good fraction of the revenues. This new market has been the subject of numerous algorithmic and game theoretic papers, and yet there are many issues, such as ensuring incentive compatibility, that are still wide open. Dealing with click fraud is another important question as is the question of selling some of the keywords via a non-auction based, announced-prices mechanism.

3 Quantum Information and Computation

Among the many astonishing aspects of quantum theory, the last-discovered is that *information* is a very different concept in quantum than in classical physical theory. The difference manifests in several ways:

1. **Quantum information and entanglement**
   The theory of noisy quantum communication channels is substantially more complicated than is that of the classical (i.e., stochastic) communication channels, whose study was initiated by Shannon. Appropriate notions of "capacity" and related questions are presently topics of study by computer scientists and physicists.

   Likewise, the notion of entanglement has no classical counterpart, and as John Bell first demonstrated in the 1960s, gives rise to non-local effects that cannot be accommodated by any classical theory. More recently these effects have come to play a central role in the computational implications of quantum mechanics.

   Bipartite and (the much more complex) multipartite entanglement, are topics of major current research efforts by computer scientists and physicists.

2. **Quantum cryptography**
   Quantum cryptography makes use of the fact that tampering with quantum information can be detected in a way that is not true of classical information. This allows for two parties to decide upon a shared key while being assured that the information is private. Thus, encrypted messages can be sent without any hardness assumptions. Such mechanisms have been built and are now commercially available.

3. **Quantum computation**
   In the 1990s it was realized that certain computational problems which appear to be intractable (require exponential time) on classical computers, can be solved efficiently (in polynomial time) on computers that can take advantage of highly entangled quantum states. These problems include some number-theoretic tasks (factoring and discrete logarithm) whose intractability is the basis for widely-deployed cryptographic protocols.

   As a consequence, both computer scientists and physicists are intensively studying several directions: whether quantum computers can solve many more hard problems efficiently; how to replace cryptographic methods with new ones that cannot be cracked by quantum computers; and most radically, whether the discoveries of the 1990s are actually an indication that quantum theory is incorrect and that the true laws of physics do not allow the very rich entanglements required. Such quantum states have never yet been constructed, and are a novel and important falsification test for quantum theory.

In order to explore these rich interactions between computer science and physics, we had two speakers at the Caltech workshop: Umesh Vazirani (Berkeley) and John Preskill (Caltech).

Computational constraints on scientific theories: insights from quantum computation Vazirani made the point that computer scientists study how the capacity of a system scales as it grows in size. In some sense the mere focus on the fact that the Hilbert space of a closed system is exponentially large, is the product of a computational perspective. Can this complexity-theoretic view, he asked, enable us to gain a deeper understanding of quantum systems?

Interestingly many of the phenomena that we associate with the mysteries of quantum mechanics are exhibited in small systems. The study of QED (quantum electrodynamics), atomic and molecular structure, and Bell states and Bell inequalities involve only a small number of particles. Even Bose-Einstein condensates, though involving many particles, are effectively low-dimensional systems. On the other hand, the laws of quantum mechanics tell us that highly entangled quantum states will arise in various large many-body systems: for example Kitaev's honeycomb lattice and topological quantum computing. Is there a way to verify the theory of quantum computing in the context of these larger, more complex systems in a similar manner to the way that single particle systems have been verified to an exquisite level of accuracy? After all, the status of a scientific theory should rest upon its ability to withstand rigorous tests.

Unfortunately, there is a difficulty in verifying the predictions of quantum mechanics for large many-body systems because it requires exponential resources to predict the outcome of the theory! Since physicists use various approximations to predict the behaviour of these systems, one can even ask whether it is quantum mechanics, or some less-theoretically-satisfying perturbation of quantum mechanics, that is being tested by experiments. Which raises the question: is there even any way of testing the validity of quantum mechanics in the regime of exponential-dimension Hilbert spaces? These tests should be subtle enough to distinguish between QM and its heuristic approximations, yet we should be able to calculate the difference in predictions!

From the physics point of view it is a bit hard to see how this could be accomplished, but this is a place where the tools of computational complexity might be of use. Here is how. One-way functions are functions that are easy to compute in one direction but are difficult to compute in the reverse direction. These functions have been of interest to computer scientists for years because of their application to cryptography. Factoring is a proposed example, since it is believed to be difficult, given q, to find two numbers x and y such that $xy = q$. However, given x and y, it is

easy to find q. We can use a quantum algorithm to compute the one-way function in the difficult direction and if the laws of quantum mechanics hold, the computation should yield the correct answer. We can compute the function in the easy direction using a classical computer to verify the results of the quantum computation. Thus, the successful implementation of quantum algorithms to solve classically-difficult problems provides a kind of verification of the laws of quantum mechanics themselves. (It at least distinguishes them from classical or approximate-quantum theories which do not have the same computational power.)

It remains an open challenge for physics as to whether we can actually implement non-trivial quantum calculations, and whether they can be used as a means of verifying the predictions of quantum mechanics.

There are other remarkable examples in which a computational view is having significant impact on our understanding of quantum mechanics. Recent work by Vidal has shown that one-dimensional quantum systems with low entanglement can be efficiently simulated by a classical computer. This result implies that entanglement is a requirement in order to achieve the full power of quantum computation. Besides the implications for quantum computation, this algorithm is a stunning example of how an information-theoretic view of quantum systems can lead to a substantially new approach to classical simulation. While algorithms have existed to determine the ground state of many classes of quantum systems, these new algorithms provide a means of simulating their evolution over time.

Another area of potential impact for computational ideas to quantum mechanics is quantum tomography, the problem of determining a quantum state from repeated preparations and measurements of the state. One of the difficulties associated with this task is that the description of a quantum state can potentially require an exponential number of parameters. Aaronson has recently suggested an approach to quantum tomography that is based on the PAC learning model developed by theoretical computer scientists. The idea is that there is a fixed distribution over a very large set of possible measurements. In a given trial, a measurement is chosen according to this distribution and the result of the measurement is recorded. After some number of trials, we would like to predict with probability at least $1-\varepsilon$ the outcome of a measurement also selected according to the distribution. Aaronson showed that this is possible to achieve with only $O(n/\mathrm{poly}(\varepsilon))$ trials. However, it remains to be seen whether there are special cases of quantum systems for which this same result can be achieved by an algorithm that is efficient both in terms of its time complexity as well as its sample complexity. Also unknown is whether these ideas can be used to learn the actual state of restricted classes of quantum states instead of just predicting the outcome of a measurement.

Quantum information and the future of physics

Preskill described quantum information science as driven by three great ideas: quantum computation, quantum cryptography and quantum error correction. We have already touched on the first two; the third has to do with the fact that since some decoherence is unavoidable in any large quantum computer, quantum computation might be practically meaningless if we could not prevent decoherence and other sources of error from destroying the complex entanglements of the computation. There has been remarkable success by researchers from computer science, coding theory and physics in coming up with error-correction schemes.

Important questions remain in all these areas. In the domain of cryptography, it would be useful to know other applications of quantum communication besides key distribution or whether there are classical cryptographic schemes that are secure against a quantum attack. In the area of

algorithm design, we still do not fully understand the power and limitations of quantum computers. The limits and capabilities of quantum error correction have significant implications for our ability to implement quantum computers. How much noise can be tolerated in a quantum computation? What are the best ways to control real quantum systems in real time under actual experimental conditions?

Turning the tables again, we can ask what quantum information theory can say about basic physics. Below we give three fundamental challenges in theoretical physics and give some indication how a computational view might be the source of useful insight:

1. Dreams of a final theory. What theory describes the fundamental constituents of matter and what are their interactions? One can go about answering these questions by asking what computational models are realized in nature. In fact, many of the deepest questions about particle physics are concerned with the information content of systems under extreme conditions. Does information escape from an evaporating black hole, and if so, how? Why is the universe classical on large scales?

2. How come the quantum? Is quantum mechanics flawed? Quantum computation is a possible way to measure whether an alternative theory is reasonable depending on whether it gives rise to reasonable computational or cryptographic power. Quantum computation may also provide a means of testing the theory. The task of building a quantum computer (or the failure to do so) may reveal an underlying flaw in the principles themselves.

3. More is different. The idea here is that in some systems the collective behavior of many particles cannot be easily predicted from knowledge of how these individual particles interact with each other. Entangled quantum many-particle systems are a perfect example of this kind of collective phenomenon. The very notion of the entanglement of a quantum state is itself an information-theoretic view of quantum systems. The classical counterpart of entanglement for qubits is correlation for classical bits which measures how much information about one bit gives you about the state of another bit. However, quantum entanglement is much richer than classical correlation because there is only one way to observe a classical bit, whereas there are many ways to measure a qubit. The quantum correlations of many qubits has the potential to encode an exponential amount of information.

The quantum entanglement exhibited or predicted in various many-particle systems is one of the beautiful mysteries of quantum systems and may provide an essential key to illuminating fundamental questions in condensed matter physics. What are the possible manifestations of many-particle quantum entanglement? Can there be a "final theory" of quantum condensed matter or are these collective phenomena inexhaustible?

As discussed above in the context of Vidal's work on one-dimensional lattices of particles, there are already examples where understanding a quantum system in terms of entanglement has lead to advances in the classical simulation of such systems. It is believed that this kind of interplay between condensed matter physics and quantum information science will continue to yield interesting and surprising results. Computationally inspired methods for describing and analyzing quantum many-body systems will deepen our understanding of exotic quantum phases of matter. Another example is in the discovery of the fractional quantum Hall effect in which electrons moving freely on a "table" at low temperature form an entangled state in which the local particle excitations are very different from those of the constituent electrons.

Meanwhile atomic physicists have developed tools for controlling and cooling atoms. Exploiting these tools we can study and discover many-particle phenomena that have been previously inac-

cessible. These experiments can be further guided by insights into quantum entanglement. This kind of interplay has already been fruitful in the study of many-body physics with polar molecules, quantized vortices in fermion pair condensates and quantum phase transitions in optical lattices.

4 Statistical Physics

Statistical physics provides a striking example of the convergence of ideas from theoretical computer science and the natural sciences. This convergence is taking place both at the level of the overall viewpoint of models and research problems (the "lens") and at the level of methods where sophisticated mathematical techniques are being developed and transferred. The domain of applications extends into many important areas, including artificial intelligence, reliable data transmission and the study of large complex networks. This section of the report is based in part on talks given at the Caltech workshop by theoretical physicists Andrea Montanari (Stanford) and Gavin Crooks (Lawrence Berkeley Laboratory).

We begin by outlining the principal conceptual elements in the interaction between statistical physics and TCS.

4.1 Common Themes

Statistical physics studies the macroscopic properties of large systems of simple components, which undergo local interactions at the microscopic level. Thus for example the properties of water are understood in terms of the interactions of H_2O molecules, and those of a magnetic material in terms of individual atomic spins. A similar situation arises frequently in computer science, e.g., when studying the global properties of large networks (such as the World Wide Web), or the structure of complex combinatorial problems described by simple constraints.

A second common theme is that random systems are often studied in order to gain insight about the behavior of large, complex (but non-random) systems. Statistical physics uses random interactions to capture the fact that many materials are heterogeneous. In computer science, randomness is used in several distinct ways: for example, the behavior of algorithms on random inputs is taken as an important benchmark for their performance; errors in unreliable devices are modeled using random noise; and many of the most successful models of real-world networks are based on random graphs that capture the salient properties of the network in a statistical sense.

Thirdly, the central concept of a phase transition in statistical physics has a close parallel in the notion of a sharp threshold in computer science. A phase transition occurs when an infinitesimal change in the parameters governing the local interactions of a physical system causes a dramatic change in its global behavior: classical examples are the transition from water to steam at 100 degrees Celsius, and the spontaneous magnetization of iron. Examples of sharp thresholds in computer science include the robustness of hardware to errors up to some critical value of the error probability on each component, and the well-known fact that many combinatorial problems suddenly switch from being "easy" to being "hard" at some particular value of a parameter describing the input.

4.2 Some Highlights

We give here just a few representative examples of potential cross-fertilization between statistical physics and TCS.

Constraint Satisfaction Problems

"Constraint satisfaction problems," in which multiple, possibly contradictory requirements are to be satisfied simultaneously, form a significant fraction of problems encountered in artificial intelligence, operations research and engineering. They also play a central role in theoretical computer science, being in a precise sense universal representatives of a huge class of combinatorial decision and optimization problems. From a statistical physics viewpoint, random instances of such problems can often be viewed as "spin glasses." Unlike the liquid to crystal transition, where materials transform from a chaotic state to a rigid periodic state, in the liquid to glass transition the material becomes rigid without any periodic structure. This corresponds to the fact that sparse instances of the constraint satisfaction problems (i.e., those with relatively few constraints) are easy to solve, while denser instances (in the "glassy phase") have clusters of stable configurations whose structures are quite different from one another. It is at the transition point from "liquid" to "glassy" that the constraint satisfaction problems become very hard to solve. Using insights from this analogy, a group of statistical physicists led by Mézard, Parisi and Zecchina recently developed an algorithmic paradigm known as "Survey Propagation," which has revolutionized our ability to solve random instances of certain constraint satisfaction problems such as Satisfiability (or SAT). The full ramifications of this breakthrough are still being worked out, and it remains a major challenge for TCS to obtain a mathematically rigorous explanation for the spectacular behavior of the resulting algorithms. More broadly, combinatorial and probabilistic insights into the algorithms will potentially lead to a better understanding of the physical spin glass models, and of the nature of ground states of complex disordered systems.

Belief Propagation, Error-Correcting Codes and Statistical Inference in Graphical Models

Survey Propagation may be viewed as an elaboration of a simpler message-passing algorithm known as "Belief Propagation," which is very widely used in such areas as coding theory and artificial intelligence. Indeed many of the most efficient codes used for reliable data transmission today rely on Belief Propagation to decode messages corrupted by a noisy channel. Despite its practical importance, the behavior of Belief Propagation is still not well understood. The statistical physics viewpoint mentioned in the previous paragraph, allied with analytical techniques developed in theoretical computer science, holds real promise for a full understanding of this important class of algorithms, and consequently for the design of powerful new error-correcting codes that are efficient in both time and bandwidth. A further ambitious goal in this direction is to make rigorous the use of Belief Propagation as a tool in statistical inference, where it is frequently used to compute the marginal probabilities in graphical models (Markov random fields). While this method is reliable for tree-like models, it is often used in more general graphs with short cycles though its behavior there is very poorly understood.

Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a classical algorithmic paradigm that is applied in many fields, from scientific computing to applied statistics to artificial intelligence. In recent years it has been exploited in novel ways in computer science to obtain efficient approximation algorithms for a number of benchmark problems, such as computing high-dimensional volumes and integrals and computing the permanent of a matrix. In statistical physics, MCMC frequently goes under the name of "Glauber dynamics," and, in addition to its algorithmic importance, it also provides a plausible model for the evolution in time of the underlying physical system: individual components

of the system update their configuration randomly based on the configurations of their neighbors. Analytical techniques developed in computer science in connection with MCMC algorithms have been successfully applied in the physical setting also. One striking phenomenon that is emerging from these investigations is that the physical concept of a phase transition often has a computational manifestation, in the form of a sudden dramatic increase in the running time of associated MCMC algorithms. Among other things, this means that the analysis of MCMC algorithms, as performed in TCS, can actually help to understand the evolution of the underlying physical system and to pinpoint phase transitions in it. In addition, the study of dynamics in networks allows us to investigate such important phenomena as the spread of computer viruses in the Internet, the evolution of viruses and epidemics in biological networks, and the propagation of influence of websites in the World Wide Web.

Non-equilibrium statistical mechanics
Most current interactions between statistical physics and TCS are based on properties of systems in equilibrium. However, a much more challenging task is to understand the non-equilibrium properties, including the approach to equilibrium and the behavior close to equilibrium states. This is relevant, for example, in designing and modeling molecular-scale engines and other computational devices. (See also the Nanotechnology section of this report.) Algorithmic methods for sampling such meta-stable states are not well-developed, though certain approaches developed by computational physicists such as "transition-path sampling" are effective in certain settings. This is an area where TCS algorithm designers, freed from the constraints imposed by physical systems, may be able to contribute novel ideas. Suitable "planted" models of constraint-satisfaction problems may serve as a useful abstraction for non-equilibrium problems.

Percolation and Sensor Networks
Novel technologies such as sensor networks and "smart dust" can be modeled as a large number of simple elements distributed randomly in space. Each element (or sensor) can detect those other sensors that are within some prescribed distance from it. This situation is reminiscent of the so-called "continuum percolation" model in statistical physics, which describes the connectivity properties of random spatial systems. Percolation models are by now very well understood; however, in applications to sensor networks there are a number of important and challenging twists. For example, the sensors are typically simple and have only very limited memory, meaning that they are able to store only a restricted amount of information about their environment. And while classical percolation is concerned only with the question of whether the system is connected, in sensor networks one is interested in more complex questions, such as the cost of routing and the ability of the network to implement certain distributed algorithms. The resolution of these issues will draw on existing insights from the theory of algorithms and percolation theory, and will extend both theories in significant ways.

4.3 Conclusion

The interaction between theoretical computer science and statistical physics is changing both areas in fundamental ways. It provides concepts, tools and algorithms that are relevant to many complex problems, including constraint satisfaction problems, the construction of efficient codes, statistical inference in artificial intelligence, and the behavior of complex networks.

5 Neuroscience

At the first workshop, Peter Dayan and Terry Sejnowski gave a *tour-d'horizon* of frontiers of neuroscience research, focusing on the growing connections between neuroscience and algorithmic thinking. One sees increasing possibilities of interaction between neuroscience, mathematical psychology (which formulates mathematical models of human and animal learning based upon behavioristic experiments) and algorithms research. This mathematical understanding of human reasoning may have other repercussions. At the second workshop, Colin Camerer and Antonio Rangel outlined a new research area, *neuroeconomics*, that seeks to inform traditional economics with behavioral models from neuroscience and psychology.

Computational models are the dominant models in neuroscience today. More sophisticated than the simpler "neural nets" studied since the 1940s, these models bear a strong resemblance to models such as Markov Decision Processes (MDPs) that are ubiquitous in machine learning and many other areas. The worldview underlying these models is Bayesian, whereby uncertainty is modeled using probabilities. Furthermore, this is not merely a modeling choice (as it is in AI): a growing body of work strongly suggests that a Bayesian vocabulary actually underlies the brain's architecture. For instance, a series of experiments culminating in the work of Schultz et. al. uses studies of dopamine levels in monkey brains to suggest that certain neural processes implement a version of *Temporal Difference (TD) learning,* a popular algorithm in reinforcement learning which at each step predicts the expected reward/punishment at the next step. In this setting, "prediction" and "reward/punishment" consist of rising/falling dopamine levels.

The above experiment is one of many recent ones that underline the importance of neurotransmitters such as dopamine, norepinephrine, and acetylcholine in neural computation, and point to the incompleteness of the older "neural net" framework. For instance, the norepinephrine system consists of neurons that make a very large number of connections (say a million, about two orders of magnitude larger than the connectivity of a typical neuron). What could they be doing that would be of interest to such a large part of the brain? They appear to be related to *vigilance,* specifically, a top-down signal that provides feedback to lower levels about the validity of recent predictions.

Further inadequacies in the traditional ideas are also clear in new evidence that timing of neural spikes is also very important, and furthermore, brain components can modulate this timing behavior (as a way of "paying attention") which causes coordinated spiking that punches through the background noise of spikes around them. Such "phase locking" can be achieved very quickly between brain regions that are quite far apart, suggesting that the underlying algorithm is simple and fast.

Many interesting questions arise for computer scientists. It would be interesting to formulate interesting new algorithms for current models for neural computation, since these models seem more powerful than the traditional neural net. A good start would be algorithms for subtasks such as data representation (e.g., using *population codes*), and modalities for sensing, inference, memory tasks, etc. Probabilistic analysis and Bayesian data models would play an important role in these algorithms. The importance of neurotransmitters suggests that continuous (as opposed to discrete) modeling may be necessary. Another interesting task would be to use computational complexity to identify limitations of these models; this was done many years ago for neural nets. In general, algorithmic issues of achieving synchronization and information transition across a noisy unreliable network seem very related to issues studied in the past two decades in computer science.

Another intriguing fact about neuroscience is that the main object being studied is (for ethical

reasons) only amenable to very limited type of "queries": a current or potential is induced on a small subset of neurons, and changes in behavior are observed. It would be interesting to step back and clearly formulate the power and limitations of this kind of queries. The fact that so much has been discovered even with such limited queries may suggest that at least many portions of the brain are not so complex.

Another promising idea is to enhance the power of the above limited "queries" using powerful algorithms (analogous to the use of algorithms in genomic sequencing). One example is the use of ICA (*independent component analysis,* the analog of PCA for nongaussian data models) in work of Sejnowski et al. that shows that the humble EEG can be analysed in a more sophisticated way than earlier thought possible. Specifically, what was considered to be "noise" in older algorithms was found using ICA to consist of data from nearby brain regions (similar to mixed up voices of conversations at a cocktail party) which provides meaningful insight into the process by which the brain as a whole "pays attention."

6 Systems Biology and Genetics

This section is in two parts. In the first part, we outline some broad challenges in molecular biology and indicate some potential roles for TCS. In the second part we discuss more specifically the talks given by Gill Bejerano, David Botstein and Dannie Durand.

6.1 A revolution in biology

The past two decades have witnessed a revolution in biology. Advances in computation and instrumentation have enabled, for the first time, a *quantitative* characterization of biological systems. This will inevitably lead to a much deeper understanding of the processes of life, and will fundamentally change the ways in which we diagnose and treat disease, including improved methods of medical diagnosis, the discovery of new drug targets, and an understanding of the action of drugs at an earlier stage of the drug development cycle.

Genomic sequencing technology has now advanced to a point at which the entire genomes of a number of model organisms, and most significantly the human, have been sequenced, and microorganisms can be routinely sequenced in a single day. As is well known, algorithmic techniques from theoretical computer science played a key role in the development of sequencing technology. (For further reading see the shotgun technique, and the human genome project at Celera Genomics.)

With these successes, attention has shifted to the task of understanding how genes and their associated proteins work in concert to regulate the processes of the cell. Although all cells within an organism contain the same genes, individual cells generate only a subset of the possible protein products of these genes, and do so only at specific times and under specific environmental conditions.

Modern molecular biology views an animal as a highly complex, precisely regulated spatial and temporal array of differential gene expression. This expression is regulated by a complex network of interactions among proteins, genomic DNA, RNA and chemicals within the cell. Technologies such as DNA microarrays, mass spectrometry, two-dimensional gel electrophoresis and *in situ* imaging are enabling the quantitative measurement of many features of this network, including gene expression, protein-protein and protein-DNA interactions and protein structure. The challenge is to organize this data into coherent models of cellular processes, which presumably will look very much like a hardwired biological computational device.

It is clear that this ambitious research agenda is inherently interdisciplinary in nature, calling for input not only from the biological, physical and engineering sciences but also from computer science. Sophisticated algorithms will play a central role in the manipulation and analysis of sequences, networks, evolutionary trees and other combinatorial objects that will arise in cellular models. TCS ideas in machine learning will be crucial in extracting structure from large biological data sets, including exploratory data analysis, pattern recognition and discovery, prediction and classification. And, at a deeper level, the interpretation of the cell as a highly complex computational device with specific functions within a larger network brings into play a whole raft of TCS modeling techniques.

We now outline a few examples of specific areas in which TCS can play an important role.

DNA Sequence Analysis

Now that the human genome and the genomes of many other species have been sequenced, extensive efforts are underway to compare these genomes, to understand how they have evolved, and to identify the genes and the associated sequences that contribute to the regulation of their expression through the processes of transcription of genomic DNA into mRNA, and translation of mRNA to protein.

A fundamental tool is the alignment of the genomes of two or more related species to identify common structure and its variation in the course of evolution. Such alignments have revealed the existence of segments that have survived virtually unchanged over millions of years of evolution. Evolutionary theory predicts that these "ultra-conserved" regions represent novel functional elements of the genome, but their exact function remains a mystery. The comparative analysis of genomes has also spawned the field of *paleo-genomics*, the reconstruction of the genomes of extinct ancestral species by tracing back from the genomes of related extant species.

Another active area is *metagenomics*, the sequencing and analysis of communities of organisms. One such community is the *human microbiome*, the collection of microbial cells occurring in or on the human body. The number of microbial cells exceeds the number of human cells by a factor of at least 100. The human microbiome plays a fundamental role both in sustaining human health and in contributing to disease. A major effort is underway to sequence the microbiomes of many individuals and study its development over the life course, its variation among groups and individuals, and how these variations affect health and disease.

Much of the information about one's risk for a disease is hiding in the genome. Geneticists use computation and statistical analysis to find individual genes that may play a role in disease and to disentangle the genetics behind complex human disorders such as cancer, diabetes, schizophrenia and obesity that are thought to arise from subtle disturbances in dozens or hundreds of genes.

Regulation of Cellular Function

A central goal of biology is to predict the behavior of cells and organisms in response to genetic and environmental changes. This response is governed principally by networks of genes, proteins and RNA molecules that act in concert to regulate cellular function.

We describe three aspects of the analysis of cellular networks: cis-regulation, protein-protein interaction and the logic of cellular pathways.

The Cis-Regulatory Code. The transcription of genes to mRNA is regulated by the binding of proteins called transcription factors to DNA in the *promoter regions* adjacent to genes. Such regulatory interactions are encoded by sequences in the DNA to which transcription factors bind. The goal of computational cis-regulatory analysis is to identify these binding sites and determine

the combinatorial control of transcription by the binding of sets of transcription factors to promoter regions. Detailed analysis of an important gene in the sea urchin has revealed that these interactions comprise a hardwired biological computational device, determining whether, and at what amplitude, each gene is expressed in each cell, throughout developmental space and time.

Protein Interaction Networks. Molecular machines within a cell can be regarded as sets of interacting proteins organized to perform a function, such as translating mRNA to protein, transporting chemicals within the cell, or orchestrating cellular response to external signals. Databases of interactions are available for many species, and their combinatorial analysis can reveal patterns of interaction among proteins that are conserved across several species. The discovery of a conserved set of densely interacting proteins can suggest that these proteins comprise a molecular machine, especially if the proteins share a common functional annotation and pattern of expression under diverse conditions.

The Logic of Cellular Pathways. A cellular pathway is a network of proteins whose levels of activation are mutually interdependent and also dependent on external stimuli. Such a network can be described by a circuit diagram akin to a logic circuit, in which the state of each wire represents the activation level of a protein or the presence or absence of an environmental stimulus, and each gate represents the deterministic or stochastic rule by which the states of incoming wires enhance or inhibit the activation of a protein. Once the proteins involved in such a pathway have been identified, the structure of the circuit can sometimes be inferred by observing how the state of the circuit changes under selected perturbations that artificially activate or deactivate a protein, or supply or remove an external stimulus. Computational learning theory is essential for modeling such circuits and adaptively choosing informative sequences of perturbations.

Synthetic Biology

Synthetic biology is concerned with inducing cells to perform new functions by embedding synthetic gene networks within them. Researchers have created a library of genetic building blocks that regulate processes such as transcription, translation and chemical modification of proteins, and have combined these parts into network structures that elicit new behaviors in a programmable fashion. One ongoing project in this area aims to program bacterial cells to become a low-cost producer of artemisinin, an important antimalarial drug. Synthetic biology is discussed in more detail in Section 7 below.

6.2 The Talks

The three biology workshop talks discussed specific examples in some of the areas discussed above, including comparative genomics, the evolution of proteins, and understanding the behavior of simple organisms such as yeast.

Insights from Comparing Genomes

Gill Bejerano's talk, "Deciphering the Human Genome: Computational Insights and Opportunities," brought out the fascinating puzzle of *ultraconserved elements*. By investigating sequenced genome data from many different species, his group discovered a collection of large segments exactly conserved (i.e., identical) in human and mouse genomes and nearly identical across all birds and mammals. The puzzling aspect is that this degree of identity is much greater than one would expect for regions that code for proteins, since this code is highly redundant and thus one would expect a much greater number of random mutations to have crept in. The question then becomes, what

alternative function might these portions of the genome have? Subsequent experiments revealed that at least some are involved in regulating when genes are expressed in embryonic development. From a CS perspective, this direction of research suggests two potential avenues for impact of computational thinking: at the algorithmic and data analysis level, one might ask what other types of correlations among genomes one can hope to efficiently discover that might lead to further insights into biological function; and at the modeling level, can an understanding of simple computational models be used to aid biologists in their search for the function of other parts of the genome?

Insights from Comparing Proteins

Dannie Durand's talk, "Trees, Graphs and the Evolution of Sequence Families," focused on the problem of inferring the evolutionary history of protein families from the profile of *domains* that the proteins contain. Large proteins consist of multiple domains, and at a high level one can describe a protein as a bit-vector indicating which domains are present or absent. One can then attempt to produce evolutionary trees for a collection of proteins by hypothesizing ancestral proteins as internal nodes such that there are not too many insertions and deletions of domains through the tree. The biology suggests different kinds of optimization criteria. For instance, one can ask for a "perfect phylogeny," which is a tree in which every bit changes state at most once anywhere in the tree. However, this is often too restrictive for real data. An alternate notion known as "Dollo Parsimony" states that insertions should happen at most once, but deletions can happen in multiple places throughout the tree. However, this provides too *little* guidance (e.g., one can always start with an ancestor that contains everything). Durand proposed instead an alternative property, "Conservative Dollo Parsimony," which requires that any *pair* of domains (bits set to 1) in an internal node A must also exist in some descendant D of A. She found that for proteins containing a moderate number of domains, while only a small fraction admitted trees satisfying perfect phylogeny, nearly all admitted trees satisfying this condition, and yet this condition would be very unlikely to be satisfied by random sets of domains of the same size (the null model). This suggests that the property is a useful guide to evolutionary tree construction. Furthermore, a tree satisfying this property, if it exists, can be found efficiently. From a TCS perspective, one interesting aspect of Durand's work is that the *algorithm* for computing a tree satisfying Conservative Dollo Parsimony is based on a *characterization* that such a tree exists if and only if a related "domain overlap graph" is chordal, and this characterization can then be explored for its biological significance. Properties of this type — that have multiple interpretations in different representations — are often important indicators of some underlying structure. TCS can potentially serve as a source of ideas in this regard in that many of the tools developed in TCS are actually techniques for moving between different representations of combinatorial objects.

Understanding Regulatory Processes

David Botstein's talk, "Metabolic Homeostasis and Growth Rate Control in Yeast: A Challenge for Data Analysis," focused on the difficult problem of understanding the interactions among different processes in an organism, and how they are affected by resource levels or stresses in the environment. For example, if yeast is placed in a "batch" environment with a limited supply of phosphate, then it will grow until the phosphate is exhausted and then will enter a stressed state in which it stops the cell cycle. On the other hand, if yeast is placed in a chemostat (a device where nutrients are supplied and culture is removed at a steady rate) with a limited rate of phosphate, then it behaves differently. In this case, the yeast will arrive at a steady state in which it limits its growth rate to adapt to the rate of introduction of phosphate and (as measured by gene expression data) will not

become stressed. From a TCS perspective, a natural question is what kind of computational or game-theoretic models can produce this type of observed behavior, and whether these can provide insight into how and why yeast acts in this manner.

7 Synthetic Biology

Combining biology with engineering, synthetic biology involves constructing biological systems with specific desirable behavior. This problem involves many core Computer Science issues: it requires designing structures that exhibit modularity, fault-tolerance, and programmability, as well as understanding how they should be composed together.

Christina Smolke's talk, "Engineering Molecular Control Systems for Programming Biological Systems" focused on the design of sensor-actuator control systems for gene expression regulation. These systems are single macromolecules that might, for instance, sense the concentration of some target (say a certain metabolite), and based on that concentration change its state to perform some desired action (such as blocking one of the steps in synthesizing an enzyme producing that metabolite). In order to be able to design a wide variety of control systems, one needs to be able to construct such molecules in a modular and standardized way. For example, Smolke's talk described how given a target of interest, one can almost mechanically design molecules that bind to this target with high specificity. One then needs to construct a communication component in the molecule so that this binding then activates an actuator that performs some desired task. Much of the engineering methodology here is very familiar to Computer Science: designing basic analog and digital components, as well as designing the "wires" that cause the output of one component to act as the input to another. The fact that the process can be largely automated introduces an important algorithmic aspect, and the constraints introduced by the biology pose novel challenges.

Adam Arkin's talk, "Signaling, Uncertainty and Design of Natural and Artificial Cellular Networks," discussed (among other things) design principles of cells and cellular networks. One can view a cell as playing a game against its environment: it needs to sense the state of its environment and be able to make the right decisions based on those sensations. In order to design cells with specific desired properties, one thus needs to program them with good strategies for this game, and in such a way that they won't mutate away from these strategies. As a driving application, Arkin discussed the problem of constructing a tumor-killing bacterium. Such a bacterium would need to sense when it is in a tumor, and then only in that environment, implement a program to invade and kill tumor cells. In addition it would need to be able to evade the body's immune system and survive in the blood long enough to find such tumors. These two properties could make it potentially dangerous if it mutated or shared genetic material with other bacteria, leading to questions of how one could design organisms with well-specified safety properties. In many respects, these design questions are reminiscent of questions in Computer Science in the areas of formal methods and distributed systems.

8 Control

Control mechanisms have long played a key role in major advances in engineering—recall Watt's steam engine governor. A control mechanism consists of a computation sandwiched between sensing and actuation. In the pioneering control mechanisms of the industrial revolution, this computation was simple, but in modern engineered systems, as well as in biology, the computations can be

complex, and their design and analysis is coming to depend upon insights from the theories of algorithms, distributed computation, and information. Moreover, in many systems there is no longer a single point of control, but rather many, communicating either implicitly through the system dynamics or explicitly through communication channels. This is the topic of distributed control, which brings new challenges of complexity, scalability and coordination and calls for yet more consideration of the constraints on communication and local computation.

The Caltech workshop featured two speakers upon these topics. Richard Murray (Caltech) spoke about "Control in an information-rich world", and Ali Jadbabaie (U Penn) spoke about "Distributed motion coordination in networked dynamic systems".

Control in an information-rich world

Murray described how classical control theory successfully achieved performance and stability goals—and tradeoffs between these—in well-understood systems such as cruise control and autopilot. The future of control theory is in complex information-rich environments such as routing and autonomous driving. Some of the key issues that come up are:

- Dynamic systems with hanging states and changing input.
- Goals are ongoing – not one-time input-output problems. Stability, performance, fuel efficiency are typical goals.
- The control mechanism needs to be fault-tolerant (for example, repeatable performance of amplifiers with 5× component variation), and in addition ensure fault-tolerance of the entire system.

Murray discussed several problems calling for insights and methods from computer science:

1. Air travel rerouting. When there is a problem in one location, people can automatically be rerouted to avoid that location. The rerouting should try to minimize the number of people that must be rerouted and may also try to satisfy (to the extent possible) a number of constraints.

   Challenges here lie in the intersection of the fields of combinatorial optimization and online algorithms.

2. Engineered biological control systems. The challenge here is that, even though isolated components can be relatively well-understood, their composition can be very complex. If one is trying, for medical purposes, to implement control mechanisms within biological systems, one may need to design algorithms with extreme fault-tolerance.

   In the medical environment there is an additional challenge: if a control system is introduced into the genetics of living organisms, it can spread by reproduction. Tracing and analyzing the complex consequences of such interventions is similar to some of the challenges due to faulty or malicious code in computer networks.

3. Control and computational insights are needed in biology not only in biological system design but also in the basic science of understanding organisms and ecologies. This agenda is generally known as "systems biology." One of the main frontiers is figuring out how the network interconnections create robust behavior from uncertain components in an uncertain

environment. This requires analyzing the system primarily from the point of view of the processing and flow of information.

4. Autonomous navigation. A navigation system combines several input *streams* consisting of massive amounts of data and must make real-time decisions. Moreover, many of the inputs may be faulty. Online machine learning may be particularly useful here.

5. Air traffic control.

6. Future fleets of coordinating autonomous automobiles or aircraft.

7. Autonomous control in deep space (time scales and communication bandwidths that preclude human intervention).

These challenges are becoming unavoidable in current heterogeneous networks that merge communications, computing, transportation, finance, utilities, manufacturing, health and entertainment.

For example: congestion control on the internet; stabilization and efficiency of power and transportation systems; the same goals in financial trading systems; and the analysis of ecosystems, including regional or global change.

From the computer science point of view there are two separate kinds of tasks here: control *over* the network, and control *of* the network. The former is primarily an engineering goal but also involves social considerations (privacy and legal ramifications of access to remote resources). The latter, while also being essentially an engineering goal, is also a scientific one, because the internet is essentially a found artifact: it is held together by a handful of simple protocols for communication and control (these are its "reductive science" laws) but it also exhibits global phenomena quite apart from any elementary rules we can describe it by. (The story is complicated of course by the fact that the network is heterogeneous, multiply-owned and not in equilibrium.) This is an important topic for scientific exploration through the "CS lens", and shares much with the statistical physics agenda described elsewhere in this report.

Distributed Motion Coordination

Jadbabaie focused in his talk on distributed control, one of the most exciting current research areas in control, and one in which the computation and communication resources available to the agents play an obviously pivotal role. Historically, control theory has studied the dynamics of a single agent, including very complex agents (such as hybrid continuous-discrete systems). Jadbabaie's talk focused on the very active area studying the dynamics of many agents, under simple (even toy) individual dynamics, laying the ground for research that will ultimately be able to analyze multiple agents with complex individual dynamics.

A particular form of collective behaviour that has drawn much attention is flocking, or schooling. This is an important and long-studied behaviour in animals, and can also be useful in robotic drones, whether aerial or in the vehicles on future automated road systems. In addition it serves as an elementary example of the emergence of global phenomena from simple, local decisions.

Most of the research in this area has been either observational (dating back to the 1950s in animals) or by simulation; very little has been done rigorously. A similar truth holds for some important widely-deployed distributed protocols (e.g., those in the 802.11 wireless standard) – we simply do not understand enough about the dynamics of most of these distributed control systems

to reliably predict how they will behave, or to entrust lives to their performance. This is a very different state of affairs than is true of other branches of engineering. It will be very valuable if the analytic techniques that have yielded proven performance guarantees in distributed computing (in the "discrete variable" and "non-real-time" setting common in CS), can be brought to bear on the distributed (and hybrid) control systems that are being implemented in our societal infrastructure.

The combination of perspectives from control, mathematics and computer science has been beneficial in some of the recent work on flocking. The original observational work on birds in the 1950s suggested individuals were copying actions of their neighbors. In Vicsek's nonholonomic model for flocking of robots (called in this context "boids"), communication and observation among boids is restricted to nearby neighbors, and boids change their orientation to an average of the previous orientations of themselves and their neighbors.

This strategy is closely related to the averaging that occurs in the heat kernel; in the discrete (graph) model in computer science, this operator (the Laplacian) plays a central role in the convergence analysis of Markov chains used in randomized algorithms. One of the key differences in the flocking scenario is that the graph of the Laplacian – i.e., the connectivity graph among nearby boids – changes over time. So one can only apply the Laplacian-type analysis if the graph stays connected "sufficiently often" over time, and if in addition, the boids are able to run a local simulation of the heat kernel that does not depend on the global structure of the graph, and that converges if the network stays connected long enough.

The ideas that go into this are many, and are due to mathematicians (dating back to work of Hajnal in the 1950s on updating "opinions" based on a nonhomogeneous Markov chain; and to the topological work of Vietoris and Rips), to computer scientists studying distributed computing (e.g., from the 70s and on, Herlihy, Lamport, Lynch, Pease, Shostak and others), to operations researchers (distributed-gradient methods of Tsitsiklis and others from the 80s), and of course to control theorists. One of the important contributions here has been a recent paper by two CS-theorists (Kempe and McSherry) giving a decentralized algorithm for spectral analysis.

An interesting issue that remains in this area is this: if one boid is "deaf", and does not average his orientation with his neighbors but continues in his own way, he will gradually bring everyone around to agree with him. This is normally not desirable, since one faulty agent has too much influence on the system. There are opportunities for useful contributions here both from the distributed computation community (the "Byzantine generals" problem and the like), and possibly even from the analysis/combinatorics/cs-theory community (the "influence of variables").

Another interesting direction that needs to be studied is what happens if each boid has a preferred orientation, so that it is not just averaging obediently with its neighbors, but partially also factoring in its own preference. And this is only the beginning of a much larger and more important investigation that is ripe for pursuit using the insights of algorithmic game theory, a theory that has recently sprung up between CS and economics: what if the boids are not fully cooperative agents, but there is also some (perhaps limited) element of competition in their interaction? Think of wireless transmitters jostling for bandwidth as they synchronize themselves to time-share access to a base station; or of pack predators at a kill.

In case the coordination system is one we get to design, the question is how to get desirable equilibria (eg in internet TCP/IP) through mechanisms that serve agents with different (and at least partially competing) interests. On top of all the complex issues this raises in algorithmic game theory, we also must take account of the on-line, real-time, possibly energy-limited environment in which the agents need to act.

Flocking is not the only simply-stated and natural model of collective dynamics. Another fascinating example is the Kuramoto model of coupled oscillators. (Each speeds up if it is slower than its neighbors and slows down if it is faster.) This model is not well understood over arbitrary networks, but Jadbabaie can prove synchronization in connected networks under certain assumptions on the coupling strengths and the natural frequencies of the different oscillators. This model is closely related to well known synchronization effects in fireflies.

In short, there are numerous issues to explore here and they seem to call upon an array of insights from computer science as well as sister subjects including economics, analysis, combinatorics, topology and statistical mechanics.

Before concluding it is worth noting that in 2002 an AFOSR panel (chaired by Murray) issued the following set of recommendations:[2] (1) Substantially increase research aimed at the integration of control, computer science, communications, and networking. (2) Substantially increase research in control at higher levels of decision making, moving toward enterprise level systems. (3) Explore high-risk, long-range applications of control to areas such as nanotechnology, quantum mechanics, electromagnetics, biology, and environmental science. (4) Maintain support for theory and interaction with mathematics, broadly interpreted. (5) Invest in new approaches to education and outreach for the dissemination of control concepts and tools to nontraditional audiences.

9 Nanotechnology

Broadly speaking, nanotechnology refers to the manipulation and control of matter on a scale of less than a micrometer (typically about 1 to 100 nanometers), and to the fabrication of devices of this size. This is a very broad, interdisciplinary field, and has connections with at least two other themes in this report, namely quantum computing and synthetic biology. In this section we focus on the part of nanotechnology that is concerned with the manufacture of electrical and mechanical devices. This topic was addressed by the talks of Nadrian Seeman (NYU), who discussed the state of the art in DNA self-assembly, and James Heath (Caltech) and Philip Kuekes (HP Labs) who presented recent work at the frontier of nanoelectronics.

9.1 Nanoelectronics and self-assembly

The field of nanoelectronics aims to construct electronic computers at the nanoscale. The potential applications are far-reaching, and include biomolecular sensors for *in vivo* detection of diseases as well as a new generation of computers that extend Moore's Law beyond the current limits of silicon integrated circuits. The two major challenges here are to design molecular switches to replace the transistor in conventional electronic circuits, and to develop associated manufacturing techniques that assemble such switches into usable computing devices. The current state of the art, as described by Heath and Kuekes, can produce, for example, memories based on a crossbar architecture with a capacity of about 160kb whose size is that of about ten white blood cells.

However, the technology is currently limited by the manufacturing process, which is both complex and unable to achieve very high densities. To go beyond this, the goal is to take much greater advantage of *molecular self-assembly*, which has the advantage of being simple and potentially able to work at much higher densities. An inevitable consequence of self-assembly will be a significantly

[2]Control in an information rich world: report of the panel on future directions in control, dynamics and systems. R. M. Murray, ed., AFOSR, 2002.

higher defect rate in the resulting components, which will need to be addressed at the algorithmic level. This ambitious program poses a number of important challenges for theoretical Computer Science.

Fault tolerance
As explained above, nanoscale computers will have a very high percentage of faulty components (perhaps 70% or more). This calls for the development of new fault-tolerant algorithms and computational paradigms that are able to detect the faulty components and/or compute robustly in their presence. The issue here is in the actual computation (logic), and the routing of communication among components, rather than in the memory, where classical techniques from error-correcting codes are already in use and effectively handle the problem. Fault-tolerance lags far behind error correction, typically achieving bounds that are far from tight, and existing techniques are not practical in this setting. Perhaps recent work in theoretical CS on fault tolerant cellular automata could be extended to the nanocircuit setting; however, any such approach will have to take into account physical constraints such as the fact that signals travel along wires at the speed of light. Recent progress in fault-tolerant techniques for quantum computation offers another potential source of techniques.

New architectures
Most current nanoelectronic devices are based on the crossbar architecture. This has the benefit of simplicity, but is not sufficiently sparse and is essentially two-dimensional. Future three-dimensional architectures will likely be based on much more complex chemical structures obtained by self-assembly. There is scope here for insights from the theory of expander graphs based on Cayley graphs (which arise from the application of simple local rules and thus may be amenable to self-assembly), combined with chemical and physical constraints, to help chemists design suitable complex molecular structures for these architectures.

Algorithmic manufacturing
The manufacturing process itself is a further area where theoretical CS can potentially make a contribution. Generally speaking, one may envisage a tradeoff between the "chemical cost" and the "computational cost" in the manufacture of nanoelectronic devices: the more time and effort one invests in the assembly, the lower the defect rate in the resulting components, and thus the less effort one needs to invest computationally in overcoming these faults (in order to deliver a usable machine). Understanding and optimizing this tradeoff is likely to be of great importance in the large-scale manufacture of nanoscale devices. While the computational costs are fairly straightforward to quantify, the same is not true of the chemical costs. Perhaps the Computer Science lens can be used to construct a formal, quantitative model for the relevant chemical processes, which can then be used to optimize the above tradeoff.

9.2 DNA Self-Assembly

Besides being the genetic code for life, DNA is a very versatile molecule for synthetic chemistry. It has a regular, well-defined structure, with the replicated helical chunk being 2nm x 3.5nm, making it a natural candidate for building nanoscale structures. In addition, base pairing along the helix allows for a great degree of control over these structures. In particular, linear DNA can first be assembled into simple motifs, which in turn can be used as the building blocks for larger self-assembled structures and crystals. A wide range of such motifs have been constructed and used in

experiments for both static self-assembly and simple prototype nanomechanical devices, including molecular machines that open and close like a clamp, walk along a track, crawl along a surface or perform logical operations. This was the topic of the talk by Nadrian Seeman.

DNA self-assembly is a very active area of experimental research, and new tools are being developed at a rapid pace. These tools are extremely rich, and often combinatorial in nature. An algorithmic approach to using these tools for performing complex bio-chemical tasks has the potential for significant impact. Indeed, it is very plausible that robust self-assembly at the molecular scale could develop into a new and very powerful engineering primitive, somewhat like the engine or the semiconductor, and would thus contribute to general scientific progress across a broad spectrum. Much of the hard work in achieving this engineering primitive will inevitably be (and is being) done by experimentalists. However, the computer science lens can play an important role by providing new algorithms, error correction techniques and analysis techniques. Indeed, some research along these lines has already been initiated by the theoretical computer science community; even more significantly, there have been successful collaborations between experimental DNA synthesis researchers and theoreticians (and these results have been published in leading theory venues). The importance of input from theoreticians is likely to increase significantly as the emphasis in the technology shifts from specific devices for focused tasks to the manufacture, on a large scale, of general-purpose devices.

Some concrete avenues for future impact here include: the development of improved quantitative mathematical models for DNA self-assembly that take into account not only theoretical aspects but also practical considerations such as errors, reaction rates, scalability etc.; a systematic study of the power and inherent limitations of the technology (within the model), using tools from complexity theory and design and analysis of algorithms; novel mathematical models for the dynamics of molecular machines; insights from the theory of graph rigidity to inform the design of rigid three-dimensional structures.

10 Astrophysics and Algorithm-Driven Science

Astrophysics

Recent advances in digital mapping of the sky, notably the Sloan Digital Sky Survey (SDSS), have dramatically changed the the way that research in astronomy and cosmology are conducted. Before SDSS, verification of a scientific hypothesis required an access to a powerful telescope, which was often a scarce and not readily available resource. Now, the vast astronomical surveys provided by SDSS are publicly available to anyone interested.

This availability of easily accessible astronomical data has enabled investigating fundamental questions about the nature of universe. These include the existence of dark energy, the nature of dark matter, and the way in which the large-scale structure of the universe has evolved over time. For example, one can investigate the role of a physical law and parameter by artificially simulating a universe subject to given laws, and checking if its statistical properties are similar to the properties of the real universe (estimated from the gathered data). The fields of Computational Cosmology and Computational Astrophysics have developed around such large-scale simulation and data analysis. These fields have made substantial contributions in recent years to our understanding of what the universe is made of, the increasing rate of expansion of the universe with its connection to dark energy, how galaxies form and evolve, and other basic cosmological questions.

Tools from Computer Science, including algorithms but also TCS concepts more broadly, have

potential to impact how models are tested and evaluated in a fundamental way. At the algorithmic level, as brought out in Andrew Connolly's presentation "Streaming the sky: The challenge for astronomy in the era of petabyte surveys," efficient data structures are crucial for performing statistical analyses on massive datasets and simulations. Connolly's team examined the clustering of galaxies in SDSS data, using it to test theories about properties of the early universe and the behavior of dark matter. Their work required use of specially-designed data structures to efficiently compute the distribution of pairwise distances and certain 3-point correlation statistics in both observed and simulated data, in order to determine how well they match up.

More broadly, however, TCS may be able to help astrophysicists decide what questions to ask of their data in determining the extent to which simulation and observation match. For example, work on large-scale properties of graphs suggests quantities such as conductance, eigenvalue gaps, and forms of metric dimension that may prove useful in this regard. The study of property testing can be viewed as a theory of what types of statistics one can hope to efficiently estimate in a massive dataset or simulation. And, ideas from pseudorandomness and learning theory suggest ways to formally discuss the extent to which a large-scale simulation fits a large-scale survey. Progress in this direction seems ideally suited to a tight-knit collaboration between astrophysics and TCS researchers, who can learn each other's languages and build on each other's strengths.

Computational learning theory

As described above, there is a change in progress in how scientific research is conducted in the data-intensive field of astrophysics. The same is true of other data-intensive scientific disciplines such as biology, as well as some topics in business, finance and national security. The field of computational learning theory, which formally studies the ability to produce complex—yet sound—generalizations from data via algorithmic means, can help scientists in a wide variety of fields in this new data-intensive era. Rather than individually proposing hypotheses and testing them on the available data, this theory may be able to help scientists automatically identify new kinds of regularities and potential hypotheses, or even determine what kinds of new data might need to be gathered next, in a rigorous and statistically justified manner.

Algorithm-driven science

The potential changes in scientific methodology in data-intensive fields, go even further than a full incorporation of the lessons of computational learning theory. instead of saying that many disciplines of science and engineering await improved techniques of data analysis, one should say that these disciplines of science need to incorporate a study of algorithmic techniques into their methodology right from the start, focusing on determining what data to collect and what format to store it in so that the elusive $O(n)$ time algorithm (say) can be designed. Likewise data-collection constraints may call for recent TCS innovations such as streaming algorithms. In general, the algorithm and mode of analysis may have to be tailored to the data, and the data collection and experimentation may have to be tailored to the algorithms that can be devised. One notable example of this trend is the historic collaboration between biologists and computer scientists during the human genome project (finished a couple of years ago), which shaved off years of work and billions of dollars from the effort.

11 Computational Models in the Social Sciences

This section is based on a talk at the Caltech workshop given by Jon Kleinberg, and outlines a research agenda at the intersection of theoretical computer science and the social sciences.

Several developments have deepened the relationship between computer science and the social sciences in recent years. Two interconnected trends that stand out in particular are the increasing availability of data-sets encoding human social interactions at unprecedented levels of scale and temporal resolution; and the rich social structure that arises from — and is increasingly designed into — large-scale information systems that support activities such as blogging, media-sharing (e.g., YouTube and Flickr), community-formation (e.g., Facebook and MySpace), and collective knowledge-creation (e.g., Wikipedia). Taken together, these developments foreshadow even greater future synergies between computing and the social sciences: algorithmic models and metaphors will be needed to explain what is taking place inside these rich social data-sets, and computer scientists will increasingly need to be aware of social-science principles in the design of applications, so as to ensure that the communities of users that grow around them do so in constructive ways.

We now discuss some of the specific opportunities in more detail, organizing this discussion around three broad classes of problems.

Dynamics of Social Processes

The spread of new ideas, technologies, opinions, fads, and rumors can be viewed as unfolding with the dynamics of an epidemic, cascading from person to person as individuals pass information to their friends and exert forms of social influence. Such processes of cascading influence have been studied via different mathematical models in a range of areas: in discrete probability and epidemiology, transmission is modeled as a stochastic phenomenon, with nodes having some probability of infecting their neighbors; in game theory and mathematical sociology, nodes make decisions about whether to accept a new idea or innovation based on the behavior of their neighbors; and in distributed computing, rules for propagation are explicitly designed so as to spread information as quickly and robustly as possible.

There are fundamental algorithmic questions surrounding all these models and the phenomena they capture; many of these questions are still not well-understood. For example, given a model of cascading influence, how can we use it to identify which nodes are truly the most "influential"? Given a social network, how can we tell if a small modification to its structure or to the strength of certain relationships will dramatically accelerate or inhibit the spread of information or influence? To what extent is the success of a new idea in one of these settings "predictable" from early observations of it? Can we use computational models such as pseudo-randomness to try quantifying empirical observations suggesting that the future dynamics of cascading phenomena may, in some precise sense, be in fact inherently unpredictable?

The Structure and Evolution of Social Networks

Algorithmic processes also provide a natural framework for expressing the phenomena that drive the growth of a social network over time: such networks are subject to forces such as preferential attachment (in which "the rich get richer") and triadic closure (in which people introduce their friends to each other, thereby closing triangles in the network). While early work on these issues identified how some of the macroscopic properties of complex networks could in principle arise from pure forms of these effects, we do not have good computational models for how these effects work together — or how one could validate, from data, a hypothesis that a particular effect is strongly

at work in a particular network at a particular point in time.

There are also fascinating challenges arising from social-network properties whose evolution seems hard to explain. For example, we know from the classic experiments of Stanley Milgram and his colleagues that real social networks tend (with caveats) to be "searchable," in that people can route messages through chains of friends to far-away strangers — given, for example, geographic and occupational descriptions of these strangers. Moreover, we have models for how a random graph embedded in such geographic and social reference frames could be searchable; and we have evidence that real social networks fit the predications of these models reasonably well. What we are completely lacking is any reasonable model for why human social networks have evolved to exhibit these properties — what are the underlying forces that cause something as amorphous and hard-to-direct as human friendships to structure themselves to support what is, in effect, a class of efficient distributed search algorithms?

The Privacy Implications of Massive Social Data

All of these questions benefit from studies of detailed data-sets encoding interactions among people, and we need to consider the privacy implications of collecting and analyzing such data. Indeed, there is a long tradition of research on privacy issues in large-scale social data-sets. However, until very recently, truly massive social data was only maintained by relatively few organizations. Now, we are in a situation where essentially every on-line transaction and piece of communication is recorded, and many companies are stockpiling data on social and economic interactions for indeterminate future purposes. One then adds to this the fact that many users of the Internet, naively or otherwise, are publicly posting large amounts of personal data that can potentially be combined with the private data held by different companies, yielding hard-to-predict breaches of privacy.

The result is a level of complexity that requires formal computational models of the potential dangers and protections. In particular, one needs formal models of the attackers that might try to breach privacy; of their computational power, limitations, and incentives; and of the means that could be employed to preserve privacy against classes of such attackers. There are already promising lines of research underway that link the fields of cryptography, randomized algorithms, and data mining in an attempt to find the capabilities and provable limitations of privacy-preserving methods for data analysis in these settings, but as with all the issues we have been discussing, there are many fundamental open questions remaining.

12 Algebra, Analysis and Combinatorics

Theoretical computer science has enjoyed a rich exchange of ideas with the classical "pure" branches of mathematics. Some of these connections are very widely known (e.g., number theory in cryptography, discrete geometry in computational geometry), but some, especially those developed over the last two decades, go very deep and indicate much more to come, though they are perhaps less widely known. We touch briefly only on three such connections, as representative examples. We unfortunately did not have time to include talks on these topics at the workshops.

(1) Algebraic constructions of expander graphs. About twenty years ago, beautiful algebraic constructions (Milman, Gabber-Galil, Lubotzky-Phillips-Sarnak) were provided for *expander graphs* which are a key object in extremal combinatorics and in computer science. Among just a few of their uses, they play an essential role in derandomization, in designing networks with

favorable connectivity properties, and in designing error correcting codes. The importance and beauty of these objects has also motivated research in algebra and analysis (in particular concerning "Property T"), and concerning the expansion properties of specific groups (notably Kassabov's breakthrough on expansion in the symmetric group).

Somewhat similarly, a recent conjecture (Moore-Russell) arising from quantum computation recently drove a breakthrough bound (Rattan-Sniady) on the characters of the symmetric group.

(2) Metric embeddings. The topic of low-distortion embeddings of one metric space in another, has been important in analysis since the work of people such as Dvoretzky and Grothendieck in the early-to-mid 20th century. Such results began to play a role in computer science in the 80s, with the work of Kahn-Kalai-Linial (who employed the Bonami-Beckner hypercontractive inequalities) on influence of variables, and Linial-London-Rabinovich (who employed and extended an embedding of Bourgain) on multicommodity flow. Since these seminal papers work in the area has expanded greatly and become a genuinely interdisciplinary field with contributions, and collaborations, between analysts and theoretical computer scientists. (See, for example, the recent Edinburgh ICMS workshop http://www.icms.org.uk/workshops/geomalg .)

(3) Pseudorandomness. The question of how random-looking deterministic objects can be, and in what sense we can call them pseudo-random, is fundamental in many areas of mathematics: statistics (theory of designs), number theory (distribution of primes, exponential sums), PDE's (regularity of wave propagation) and combinatorics (Ramsey theory, discrepancy theory). In theoretical computer science, a computational theory of pseudorandomness was developed beginning in the 1980s, with applications to probabilistic algorithms, cryptography, computational complexity and weak random sources. Many connections between the above frameworks for pseudorandomness already exist, and some recent ones are particularly striking: use of the sum-product theorem to build randomness extractors, and from there to construct Ramsey graphs; use of Szemerédi's Regularity lemma in property testing and in finding long arithmetic progressions in primes; and the use of Gowers uniformity in circuit lower bounds and pseudorandom generators.

Acknowledgments

We thank first and foremost the workshop speakers for providing the inspiration and technical input on which this report is based. Many thanks also to the other invited participants, who contributed to stimulating discussions during the workshops and also provided scribe notes from the talks. In particular, we wish to thank Dimitris Achlioptas, Boaz Barak, Bernard Chazelle, Moses Charikar, Ashish Goel, Piotr Indyk, Sandy Irani, Adam Kalai, Dick Karp, Elchanan Mossel, R. Ravi, Sara Robinson, Mike Saks, Rob Schapire, Rocco Servedio, Luca Trevisan, Chris Umans and Avi Wigderson for their input, and Jon Kleinberg for writing the section on Algorithmic Models in the Social Sciences.

A The Princeton Workshop

The workshop web-site (containing videos and slides from the talks) is at:
http://www.cs.caltech.edu/∼schulman/Workshops/CS-Lens-1/cs-lens-1.html

Talks (in the order they were given)

- Leslie Valiant (Harvard): Examples of computational models for neuroscience and evolution.
- Peter Dayan (University College London): Computational neuromodulation.
- Adam Arkin (Berkeley): Signaling, uncertainty and design of natural and artificial cellular networks.
- Ned Seeman (NYU): DNA: not merely the secret of life.
- Christina Smolke (Caltech): Engineering molecular control systems for programming biological systems.
- Lou Gross (U Tennessee): Synchrony, parallelism and multimodeling.
- James Heath (Caltech): The current status and future opportunities for ultra-dense nanoelectronic circuitry.
- Phil Kuekes (HP Labs): Nanoscale self-assembly – a computational view.
- Terry Sejnowski (Salk Institute): In search of the brain's independent components.
- Gill Bejerano (Stanford): Deciphering the human genome - computational insights and opportunities.
- Dannie Durand (CMU): Graphs, trees and the evolution of gene families.
- David Botstein (Princeton): Metabolic homeostasis and growth rate control in yeast: a challenge for data analysis.

Attendees

Dimitris Achlioptas (UC Santa Cruz), Adam Arkin (Berkeley), Sanjeev Arora (Princeton), Boaz Barak (Princeton), Gill Bejerano (Stanford), Avrim Blum (CMU), David Botstein (Princeton), Moses Charikar (Princeton), Bernard Chazelle (Princeton), Peter Dayan (University College London), Dannie Durand (CMU), Martin Farach (Rutgers), Mike Foster (NSF), Ashish Goel (Stanford), Michel Goemans (MIT), Lou Gross (Tennessee), James Heath (Caltech), Richard Karp (Berkeley), Phil Kuekes (HP Labs), Stanley Leibler (Rockefeller), Adi Livnat (Princeton), Elchanan Mossel (Berkeley), R. Ravi (CMU), Michael Saks (Rutgers), Rob Schapire (Princeton), Leonard Schulman (Caltech), Ned Seeman (NYU), Terry Sejnowski (Salk Institute), Rocco Servedio (Columbia), Alistair Sinclair (Berkeley), Mona Singh (Princeton), Christina Smolke (Caltech), William Steiger (NSF), Olga Troyanskaya (Princeton), Leslie Valiant (Harvard), Vijay Vazirani (Georgia Tech), Avi Wigderson (IAS), Angela Yu (Princeton), Wei Zhao (NSF), Manfred Zorn (NSF).

B The Caltech Workshop

The workshop web-site (containing videos and slides from the talks) is at:
http://www.cs.caltech.edu/∼schulman/Workshops/CS-Lens-2/cs-lens-2.html

Talks (in the order they were given)

- Jon Kleinberg (Cornell): Algorithmic models for social network phenomena,
- John Preskill (Caltech): Quantum information and the future of physics,
- Umesh Vazirani (Berkeley): Computational constraints on scientific theories: insights from quantum computing,
- Richard Murray (Caltech): Control in an information-rich world,
- Ali Jadbabaie (U Penn): Distributed motion coordination in multi-agent systems: From flocking and synchronization to coverage verification in sensor networks,
- Andrei Broder (Yahoo! Research): Technical challenges in web advertising,
- Andrew Connolly (Pittsburgh): Streaming the sky: The challenge for astronomy in the era of petabyte surveys,
- Andrea Montanari (Stanford): Phase transitions in large graphical models: from physics to information theory and computer science
- Gavin Crooks (Berkeley and LBL): Importance sampling of trajectories in complex systems

Attendees

Sanjeev Arora (Princeton), Yair Bartal (Hebrew U), Avrim Blum (CMU), Andrei Broder (Yahoo! Research), Colin Camerer (Caltech), Tony Chan (NSF), Andrew Connolly (Pittsburgh), Gavin Crooks (Berkeley and LBL), George Djorgovski (Caltech), John Doyle (Caltech), Federico Echenique (Caltech), Michael Foster (NSF), Fan Chung Graham (UCSD), Ron Graham (UCSD), Russell Impagliazzo (UCSD), Piotr Indyk (MIT), Sandy Irani (UCI), Sergei Izmalkov (MIT), Ali Jadbabaie (U Penn), Adam Kalai (Georgia Tech), Ehud Kalai (Northwestern), Richard Karp (Berkeley), David Kempe (USC), Jon Kleinberg (Cornell), Phil Kuekes (HP), John Ledyard (Caltech), Richard Lipton (Georgia Tech), Mat McCubbins (UCSD), Adam Meyerson (UCLA), Andrea Montanari (Stanford), Elchanan Mossel (Berkeley), Kamesh Munagala (Duke), Richard Murray (Caltech), Andrew Odlyzko (UMN), Rafail Ostrovsky (UCLA), Ramamohan Paturi (UCSD), Andrew Postlewaite (U Penn), John Preskill (Caltech), Antonio Rangel (Caltech), Sara Robinson (Berkeley), Tim Roughgarden (Stanford), Eyal Rozenman (Caltech), Amit Sahai (UCLA), Leonard Schulman (Caltech), Alistair Sinclair (Berkeley), William Steiger (NSF), Luca Trevisan (Berkeley), Chris Umans (Caltech), Umesh Vazirani (Berkeley), Vijay Vazirani (Georgia Tech), Erik Winfree (Caltech), Neal Young (UCR).