2020

SCIENCE

“Science and everyday life cannot and should not be separated”

Rosalind Franklin (1920-1958)

James Watson, Francis Crick and Maurice Wilkins shared the 1962 Nobel Prize in Physiology or Medicine for the discovery of the double-helix structure of DNA. However, the discovery would not have been possible without the brilliant but short-lived Rosalind Franklin, whose work underpinned that of those above, but was never properly credited.

TOWARDS

2020

SCIENCE

Contents

This report contains the initial findings and conclusions of a group of internationally distinguished scientists who met over an intense three days in July 2005 to debate and consider the role and future of science over the next 14 years towards 2020, and in particular the importance and impact of computing and computer science on science towards 2020.

About this Report

Fourteen years is a long time to look into the future. That is why this report is *not* about attempting to predict or 'forecast' it. Instead, our starting point was simply to consider what we (and most) believe are some of the greatest challenges and opportunities for the world in the 21st century that urgently require advances in science to address. From there, we considered how computing and computer science needs to, and can, play a vital role in realising such advances, starting from how even current applications of computing and computer science are already having an important impact on science and consequently on society. Finally, we considered what needs to happen in computing and computer science – as well as in science policy and in education – to accelerate advances in the sciences that can then help address key global challenges towards 2020.

This report is the product of this meeting, together with a further four months of analysis, discussion and debate by the 2020 Science Group and others we consulted. We have striven to produce a vision of science towards 2020 and the role that computer science can and will play in achieving this vision. While each section is written solely by its named authors and reflects their view, the overall vision is one shared by the entire 2020 Group.

Surprisingly, or perhaps not given the effort involved in producing it, this report is, to our knowledge, the first to articulate a comprehensive vision of science towards 2020, the impact of the convergence of computer science and the other sciences, and to identify specifically what the science community and policy makers can do to ensure the vision we outline becomes a reality.

Our hopes for this report are that it serves three purposes. First, to help stimulate debate and discussion in the science community about the direction science might take over the coming two decades, and the increasing role and impact of computing and computer science in the sciences. We hope that such discussion will help refine and shape the issues we highlight in this report, as well as perhaps also highlighting important issues we have not considered. Second, to input into and inform science policy thinking, and in particular to underpin the importance of science in society, the changing nature of basic science and the urgent need to move the agenda beyond the currently limiting 'e-science' and 'computational science' focus. Linked to this, we would also like this report to help inform the education policy debate, especially the vital importance of ensuring that today's children can become tomorrow's 'new kinds' of scientists required to tackle key scientific and social challenges and opportunities in the first half of the 21st Century. And third, to serve towards galvanising the computer science and science communities into working more closely together in a directed, fruitful way bringing together the 'computational thinking' that underpins computer science and the empirical and theoretical methods that underpin the physical and biological sciences.

The report's emphasis is on the role and impact of computing and computer science in science. It does not focus on other developments that are also

influencing science, notably novel mathematical and statistical techniques. This is not to deny their importance. We deliberately chose to focus on the intersection of computation and the sciences because this, we argue, is the most important development for the future of science towards 2020.

The report also focuses largely, although not exclusively, on the natural sciences rather than the physical sciences, engineering or social sciences. In particular, it focuses on the biological sciences broadly defined, from molecular biology to systems biology to organismic biology and ecosystems science. The reasons are twofold. First, because it is in the natural sciences where the 2020 Group argue the greatest impact of computer science will be felt. And second, because it is in these areas where many of the greatest scientific, social and global challenges are to be found. We do not separate out 'nanotechnology' specifically, although there is of course much attention being paid to this area. Instead, we outline in several parts of the report how nanoscience technologies and applications are emerging in medicine, biology and computing as a consequence of the convergence of biology, chemistry, physics and computer science.

The 2020 Science Group is composed of over 30 scientists spanning biology, physics, chemistry, biochemistry, astronomy, genetics, medicine, mathematics and computer science, and 12 different nationalities. Coming from some of the world's leading research institutions and companies, the scientists were elected for their expertise in a particular field.

The Venice workshop which took place in July 2005, and which formed the beginning of this project, comprised a proven, structured roadmapping technique developed by Dr Robert Phaal of Cambridge University, together with 'open' brainstorming sessions to begin to define our roadmap and vision. Subsequent to the Venice workshop, the entire group has spent considerable effort in working together to develop and strengthen the scientific and technological areas and positions, guided by the 2020 Steering Board, and informed by additional outside consultations from other experts in the field and in the areas of economics and science policy.

This is our initial report. We will be refining it, and particularly the roadmap to 2020 through feedback and discussions generated by this initial report with peers, others in the science community, and with policy makers. Your contribution to this, whether it is to build upon what we have done or constructively criticise it, will be valuable in the process of making concrete an ambitious, bold but realisable vision of the aspirations of science towards 2020.

Stephen Emmott & Stuart Rison

Contacting us

We welcome feedback on the report. Feedback on any specific section should be addressed to its corresponding author. All other comments and feedback should be addressed to:

Stephen Emmott

Microsoft Research
7 J J Thomson Avenue
Cambridge, CB3 0FB
UK

semmott@microsoft.com

The 2020 Science Group

The 2020 Science Group (Venice, July 2005).

Standing (l to r): Peter Buneman, Stephen Emmott, Malcolm Young, David Searls, Andy Parker, James Maxwell Wilkinson (rapporteur), Jorge Soberon, Alex Szalay, Timo Hannay, Tetsuya Sato, René Brun, José Blakeley, Michael Franklin, Marcel Dissel (facilitator), Don Syme, Andrew Phillips (rapporteur), Andre Hagehülsmann, Neil Ferguson, Vassily Lyutsarev, Jamie Shiers, Robert Phaal (facilitator), Wolfgang Emmerich, Klaus-Peter Zauner, Simon Cox, Damien Watkins.

Sitting (l to r): Serge Abiteboul, Søren Brunak, Helen Parkinson (rapporteur), Parviz Moin, Clemens Szyperski, Manuel Peitsch, Luca Cardelli, Miroslav Radman, Ehud Shapiro, Chris Bishop, Aron Kuppermann, Stephen Muggleton, Andrew Herbert, Peter Landshoff, Anthony Finkelstein, Angela Still (administration).

Chairman

Professor Stephen Emmott
Microsoft Research Cambridge, UK

Co-ordinator

Dr Stuart Rison
Computational Sciences Group, Microsoft Research Cambridge, UK

Professor Serge Abiteboul
INRIA-Futurs, France

Professor Christopher Bishop
Head of Machine Learning & Perception, Microsoft Research Cambridge, UK

Dr José Blakeley
Software Architect, Microsoft Corporation, USA

Dr René Brun
CERN, Switzerland

Professor Søren Brunak
Director, Center for Biological Sequence Analysis, Technical University of Denmark, Denmark

Professor Peter Buneman
Professor of Informatics, University of Edinburgh, UK

Dr Luca Cardelli
Head of Programming Principles & Tools, Microsoft Research Cambridge, UK

Professor Simon Cox
Professor of Computational Methods, University of Southampton, UK

Professor Wolfgang Emmerich
Professor of Distributed Computing, University College London, UK

Professor Neil Ferguson
Professor of Mathematical Biology, Imperial College London, UK

Professor Anthony Finkelstein
Professor of Software Systems Engineering, University College London, UK

Professor Michael Franklin
Professor and Vice Chair for Computer Science, University of California, Berkeley, USA

Dr Timo Hannay
Director of Web Publishing, Nature Publishing Group, UK

Dr Andrew Herbert
Managing Director, Microsoft Research Cambridge, UK

Professor Aron Kuppermann
Professor of Chemical Physics, California Institute of Technology, USA

Professor Peter Landshoff
Director of Research, Cambridge-MIT Institute, Cambridge, UK

Professor Parviz Moin
Professor of Mechanical Engineering, Stanford University, USA

Professor Stephen Muggleton
Head of Computational Bioinformatics Laboratory, Imperial College London, UK

Professor M Andy Parker
High Energy Physics Group, Cavendish Laboratory, Cambridge University, UK

Professor Manuel Peitsch
Global Head of Informatics and Knowledge Management, Novartis Institutes of Biomedical Research, Switzerland

Professor Miroslav Radman
Director, Medical and Evolutionary Molecular Genetics, INSERM
Faculté de Médecine – Necker, University of Paris, France

Professor Tetsuya Sato
Director-General of the Earth Simulator Center, Tokyo, Japan

Dr David Searls
Senior Vice-President, Worldwide Bioinformatics, GlaxoSmithKline, USA

Professor Ehud Shapiro
Department of Computer Science & Applied Mathematics and Department of Biological Chemistry, Weizmann Institute of Science. Israel

Dr Jamie Shiers
CERN, Switzerland

Dr Jorge Soberon
Natural History Museum and Biodiversity Research Center, Kansas University, USA

Dr Don Syme
Researcher, Microsoft Research Cambridge, UK

Professor Alexander Szalay
Department of Physics and Astronomy, John Hopkins University, USA

Professor Clemens Szyperski
Software Architect, Microsoft Corporation, USA

Dr Damien Watkins
Microsoft Research Cambridge, UK

Professor Malcolm Young
Pro-Vice-Chancellor, Newcastle University, UK

Dr Klaus-Peter Zauner
School of Electronics and Computer Science, University of Southampton, UK

Contributors

Dr Brian Beckman
Software Architect, Microsoft Corporation, USA

Dr Andre Hagehülsmann
Computational Sciences Group, Microsoft Research Cambridge, UK

Professor David Harel
William Sussman Professorial Chair, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Israel

Dr Vassily Lyutsarev
Computational Sciences Group, Microsoft Research Cambridge, UK

Professor Ben Martin
SPRU, Science and Technology Policy Research, University of Sussex, UK

Dr Andrew Phillips
Computational Sciences Group, Microsoft Research Cambridge, UK

Dr A Robin Wallace
Head of Institute for Energy Systems, University of Edinburgh, UK

Summary

We present the findings of an internationally respected group of scientists who, in July 2005, all met to discuss, debate and consider the future of science towards 2020, and in particular the role and impact of computing and computer science on the sciences. This group has produced seven main findings:

1 An important development in science is occurring at the intersection of computer science and the sciences that has the potential to have a profound impact on science. It is a leap from the application of computing to support scientists to 'do' science (i.e. 'computational science') to the integration of *computer science concepts, tools and theorems* into the very fabric of science. While on the face of it, this change may seem subtle, we believe it to be fundamental to science and the way science is practiced. Indeed, we believe this development represents the foundations of a new revolution in science.

2 Conceptual and technological tools developed within computer science are, for the first time, starting to have wide-ranging applications outside the subject in which they originated, especially in sciences investigating complex systems, most notably in biology and chemistry. Indeed, we believe computer science is poised to become as fundamental to biology as mathematics has become to physics. We postulate this because there is a growing awareness among biologists that to understand cells and cellular systems requires viewing them as information processing systems, as evidenced by the fundamental similarity between molecular machines of the living cell and computational automata, and by the natural fit between computer process algebras and biological signalling and between computational logical circuits and regulatory systems in the cell. We believe this is a potential starting point for fundamental new developments in biology, biotechnology and medicine.

3 We believe that computer science concepts and tools in science form a third, and vital component of enabling a 'golden triangle' to be formed with novel mathematical and statistical techniques in science, and scientific computing platforms and applications integrated into experimental and theoretical science. This combination is likely to accelerate key breakthroughs in science and benefits to society, from understanding biology and revolutionising medicine and healthcare, and from understanding the universe to the origin of life, and understanding and helping to protect the life-support systems of Earth on which we all depend for our survival.

4 We highlight that an immediate and important challenge is that of end-to-end scientific data management, from data acquisition and data integration, to data treatment, provenance and persistence. But importantly, our findings urgently require us to reconsider current thinking in the increasingly prominent domain of 'computational science'. While advances in computing, and in particular scientific data management and application development environments for science will be important towards 2020, we believe that vitally more important, and dramatic in its impact, will be the integration of new conceptual and technological tools from computer science into the sciences. Computer science concepts provide levels of abstraction allowing scientists from different fields to understand and learn from each other's solutions, and ultimately for scientists to acquire a set of widely applicable complex problem solving capabilities, based on the use of a generic computational environment, in the same way that they learn universally applicable mathematical skills. We believe that the current view of 'computational science' as a separate 'third pillar' in science alongside experimental and theoretical science is an intermediate, unsustainable and undesirable state.

5 Our findings have significant implications for scientific publishing, where we believe that even near-term developments in the computing infrastructure for science which links data, knowledge and scientists will lead to a transformation of the scientific communication paradigm.

6 We also believe this development is not only a potential starting point for fundamental new developments in biology, biotechnology and medicine, but also for potentially profound developments in the future of computing. Big challenges for future computing systems have elegant analogies and solutions in biology, such as the development and evolution of complex systems, resilience and fault tolerance, and adaptation and learning. New levels of understanding and knowledge about biological processes and systems could underpin the new building blocks of the next century of computing.

7 Finally, our findings have significant implications for the education of tomorrow's scientists and science policy and funding. Scientists will need to be completely computationally and mathematically literate, and by 2020, it will simply not be possible to do science without such literacy. This therefore has important implications for education policy right now. The output of computer scientists today barely meets the needs of the public and industrial computing sectors, let alone those required for future science sectors. These developments will also fundamentally affect how science needs to be funded, what science is funded, and many current assumptions underpinning existing science policies. They also have economic implications. We are starting to give birth to 'new kinds' of science and possibly a new economic era of 'science-based innovation' that could create new kinds of high-tech sectors that we can barely imagine today, just as we could hardly have imagined today's rapidly growing 'genomics' sector happening two decades ago.

We outline here a *vision* for science towards 2020, and how this vision can underpin fundamental breakthroughs in science and provide benefits to societies around the world. Our vision and our findings culminate in what we understand to be the first ever comprehensive attempt to define a *roadmap* towards 2020 science, which we hope will stimulate discussion and debate and give direction for scientists, policy makers and governments, as well as inspire a generation of today's children to become tomorrow's scientists.

The 2020 Science Group

Dendritic cell and lymphocyte, coloured scanning electron micrograph (SEM)

A coloured scanning electron micrograph showing the interaction between a dendritic cell (blue) and a T lymphocyte (pink), two components of the body's immune system. Both are types of white blood cell. T lymphocytes recognise a specific site on the surface of pathogens or foreign objects (antigens), bind to it, and produce antibodies or cells to eliminate that antigen. Dendritic cells are antigen-presenting cells (APCs); they present antigens to T lymphocytes, which can only recognise antigens when they are presented by APCs.

Dr Olivier Schwartz / SCIENCE PHOTO LIBRARY

Introduction

A scientific revolution is just beginning. It has the potential to create an era of *science-based* innovation that could completely eclipse the last half century of *technology-based* innovation; and with it, a new wave of global social, technological and economic growth.

The basis for this revolution is the emergence of new *conceptual* and *technological* tools from computer science – tools which are already proving their potential to have a profound impact on science. I distinguish computer science from computing. Computers have played an increasingly important role in science for 50- years, and in particular the past decade and a half, and will continue to do so. However, what this report uncovers, for the first time, is a fundamentally important shift from *computers* supporting scientists to 'do' traditional science to *computer science* becoming embedded into the very fabric of science and how science is done, creating what I am prepared to go so far as to call 'new kinds' of science[1].

Scientific revolutions are rare, but history shows they occur when either a fundamentally important new 'conceptual' tool (e.g. calculus) or 'technological' tool (e.g. the telescope) is invented that leads to the creation of *'new kinds'* of science.

In 1202, Leonardo of Pisa (whom we now know as Fibonacci) published *Liber Abaci*, which set out a new branch of mathematics: algebra (from the Arabic *al-jebr* 'the science of restoring what is missing and equating like for like'). Algebra enabled a fundamental shift from written to symbolic mathematics - mathematics in Europe was written in words up to that point. Fibonacci 'discovered' the numerical system (the system we use today) which originated in India around 300AD, and made its way, via the Muslim world, to Europe. Algebra enabled *'computors'* (i.e. human calculators) to perform new kinds of calculations that changed society, from transforming study of the planets, to having a fundamental impact on religion and commerce. Some 400 years later, Newton, in his efforts to understand the natural laws of the rate of change in motion, used algebra to underpin another new branch of mathematics: calculus (a branch for which von Leibniz is simultaneously and independently credited). Calculus spurred scientists "to go off looking for other laws of nature that could explain natural phenomenon in terms of rates of change and found them by the bucketful - heat, sound, light, fluid dynamics, electricity and magnetism" [2]. Similarly, the invention of new *technological* tools in science, such as the invention of the telescope in 1604 by Galileo, the microscope and the discovery of X-rays, transformed science and our understanding of the world and our universe.

The developments in science under way now and highlighted in this report are likely to prove at least as important as those that had a transforming effect on science and society in the past. As a consequence of the vision we describe here, it is clear that science has the potential to have an unprecedented impact on our world in the 21st Century, from how long we live, to how we live, to what we know about ourselves, our planet and the universe, to understanding how to control and eradicate disease, to how to protect the entire life-support systems of the earth. As a consequence, it is difficult to overestimate how profound is the scientific revolution now under way.

Stephen Emmott

[1] I distinguish between the 'new kinds' of science I talk about and Wolfram's 'new kind of science' [1]. In his thesis, 'A new kind of science', Wolfram describes largely an exposition of interactions in cellular automata. Notwithstanding this, Wolfram's claim – that science will be transformed by new generalisable rules that are executable in a machine (i.e. the computationalisation and codification of science) – is in agreement with our findings.

1 Laying the Ground

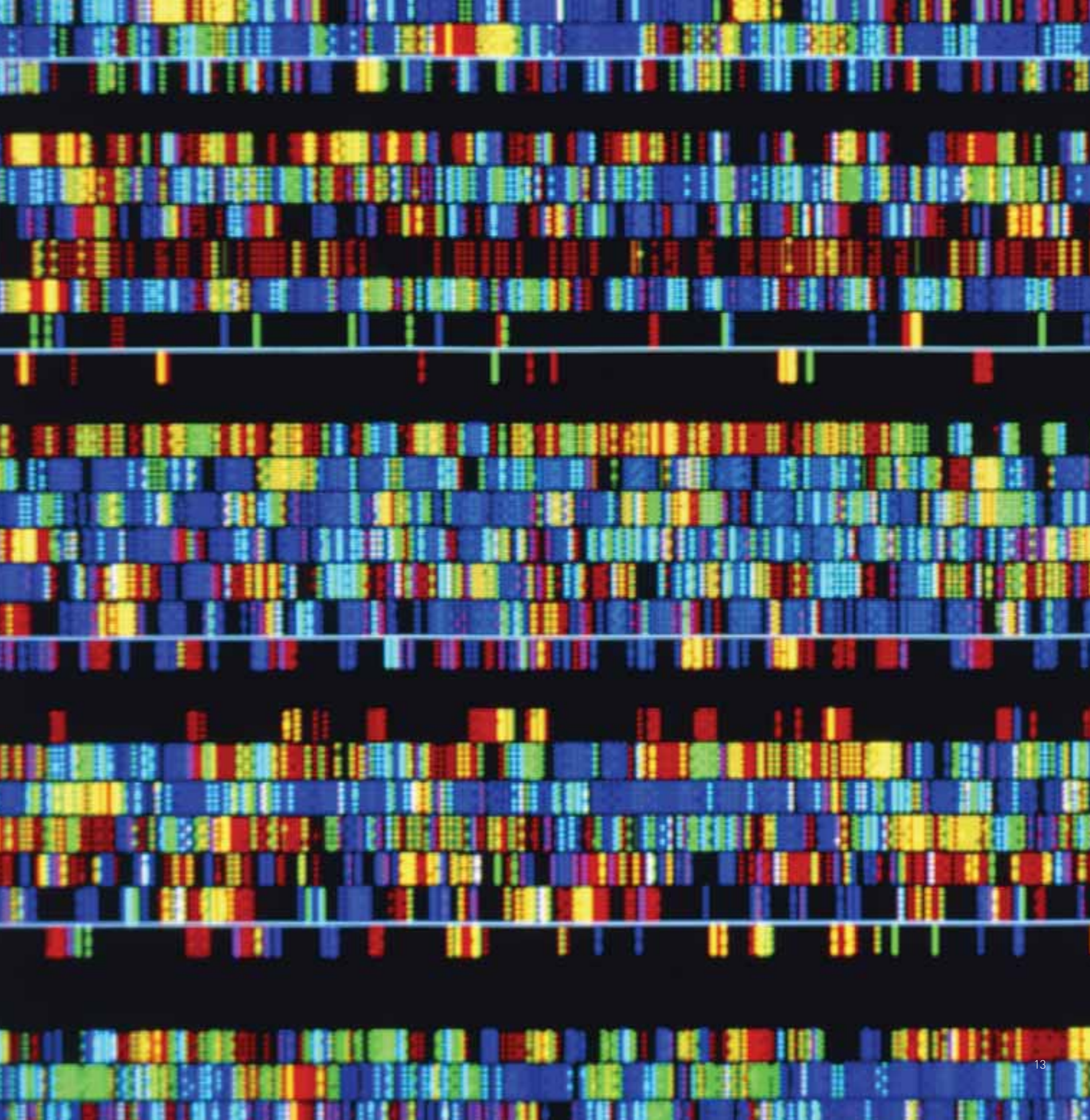
Computing has rapidly established itself as essential and important to many branches of science, to the point where ‘computational science’ is a commonly used term. Indeed, the application and importance of computing is set to grow dramatically across almost all the sciences towards 2020. Computing has started to change how science is done, enabling new scientific advances through enabling new kinds of experiments. These experiments are also generating new kinds of data – of increasingly exponential complexity and volume. Achieving the goal of being able to use, exploit and share these data most effectively is a huge challenge.

In Part 1, we consider trends and developments already under way in computing and computer science, and additional requirements needed to achieve this aim. These will lay the ground for a far more fundamental impact on science, which we cover in Part 2.

Human DNA sequence

Computer screen display of a human DNA (deoxyribonucleic acid) sequence as a series of coloured bands. This is for the human genome project. DNA consists of two long strands linked by the interactions of bases along their lengths. Each colour represents a specific base. The sequence of bases makes up the genetic code in the form of genes, segments of DNA which have specific functions within an organism. By studying the genes in human DNA, a greater understanding of genetic diseases and heredity can be achieved. Photographed at the Sanger Centre in Cambridge, UK.

James King-Holmes / SCIENCE PHOTO LIBRARY



Computational Science

Science is continuously pushing the limits of what is possible in computing, and in some areas is leading computational advances. Climate and earth system research, fluid dynamics, genomics, proteomics, theoretical chemistry, astrophysics, nanostructure physics and high-energy physics are all examples. Scientific computing platforms and infrastructures are making possible new kinds of experiments that would have been impossible to conduct only 10 years ago, changing the way scientists do science.

They are experiments that are also generating vast volumes of data. For example, The Sanger Centre at Cambridge currently hosts 150 terabytes (150 trillion [10^{12}] bytes) of unique genomic data and has a cumulative installed processing power (in computer clusters) of around 2.5 teraflops. Its genome sequence data is doubling each year, significantly faster than Moore's Law (describing the growth in computer power) [3]. Future possibilities of determining the entire DNA sequence of human individuals may keep the exponential growth going for many years. Sanger is considering new technologies such as MAID (Massive Array of Idle Disks) to see if it can keep up with this rate of data growth. And particle physics is set to generate several petabytes (a million billion [10^{15}] bytes) of data every year in the coming decade and beyond when the Large Hadron Collider (LHC) starts operating at CERN in 2007 (see the section 'Understanding the Universe' in Part 3). The analysis of the LHC data constitutes one of the greatest current challenges to scientific computing. CERN's planned solution is to use computing 'grids' and these are currently being deployed in Europe (LCG), Scandinavia (NorduGrid) and the US (Grid3) as well as in collaborating institutes in Asia and Australia.

The LCG 'vision' is one being replicated across the world with funding for huge infrastructure projects like e-Infrastructures (EU), Cyber-infrastructure (USA), Gloriad and others. However, it is important to note that the functionality offered by the current LCG has been scaled back significantly with respect to the 'Grid vision'.

Even with the relatively simple data structure of particle physics, *data management* is a major issue. It is necessary to merge the capabilities of a file system to store and transmit bulk data from experiments, with logical organisation of files into indexed data collections, allowing efficient query and analytical operations. It is also necessary to incorporate extensive metadata describing each experiment and the data it produced. Rather than flat files traditionally used in scientific data processing, the full power of relational databases is needed to allow effective interactions with the data, and an interface which can be exploited by the extensive scientific toolkits available, for purposes such as visualisation and plotting.

Disciplines other than particle physics require support for much more diverse types of tasks than we find in the large, very coherent and stable LHC 'gridded' virtual organisations. Astronomy, for example, has far more emphasis on the collation and curation of federated datasets held at disparate sites. There is less massive computation, and large-scale modelling is generally done on departmental high performance computing (HPC) facilities. Chemistry also has problems which are very different from those in particle physics. The community is formed of very small teams and relatively undeveloped computational infrastructure. In the life sciences, the problems are far more related to heterogeneous, dispersed data rather than computation.

The harder problem for the future is heterogeneity, of platforms, data and applications, rather than simply the scale of the deployed resources. The goal should be to allow scientists to 'look at' the data easily, wherever it may be, with sufficient processing power for any desired algorithm to process it. Current platforms require the scientists to overcome computing barriers between them and the data.

Next Decade

Effect of multi-core CPUs

We postulate that most aspects of computing will see exponential growth in bandwidth but sub-linear or no improvements at all in latency. Moore's Law will continue to deliver exponential increases in memory size but the speed with which data can be transferred between memory and CPUs will remain more or less constant and marginal improvements can only be made through advances in caching technology. Likewise, Moore's law will allow the creation of parallel computing capabilities on single chips by packing multiple CPU cores onto it, but the clock speed that determines the speed of computation is constrained to remain below 5 GHz by a 'thermal wall'. Networking bandwidth will continue to grow exponentially but we are approaching the speed of light as a floor for latency of network packet delivery. We will continue to see exponential growth in disk capacity but the speed with which disks rotate and heads move, factors which determine latency of data transfer, will grow sub-linearly at best, or more likely remain constant.

Thus commodity machines will not get much faster. But they will have the parallel computing power and storage capacity that we used to only get from specialist hardware. As a result, smaller numbers of supercomputers will be built but at even higher cost. In fact, this trend has started with the National Science Foundation significantly reducing the funding of US supercomputer centres [4]. From an application development point of view, this will require a fundamental paradigm shift from the currently prevailing sequential or parallel programming approach in scientific applications to a mix of parallel and distributed programming that builds programs that exploit low latency in multi core CPUs but are explicitly designed to cope with high latency whenever the task at hand requires more computational resources than can be provided by a single machine.

Commodity machines can be networked into clusters or grids of clusters and perform tasks that were traditionally restricted to supercomputers at a fraction of the cost. A consequence of building grids over wide-area networks and across

organisational boundaries together with the lack of further improvement in network latency means that the currently prevailing synchronous approach to distributed programming, for example, using remote procedure call primitives, will have to be replaced with a fundamentally more delay-tolerant and failure-resilient asynchronous programming approach. A first step in that direction is *peer-to-peer* and *service-oriented architectures* that have emerged and support reuse of both functionality and data in cross-organisational distributed computing settings.

Peer-to-peer and 'service-oriented' architectures

Peer-to-peer (P2P) architectures support the construction of distributed systems without any centralised control or hierarchical organisation [5]. These architectures have been successfully used to support file sharing most notably of multi-media files. We expect that computational science applications will increasingly use P2P architectures and protocols to achieve scalable and reliable location and exchange of scientific data and software in a decentralised manner.

While P2P systems support reuse of data, the paradigm of service-oriented architectures (SOA) and the web-service infrastructures [6] that assist in their implementation facilitate reuse of functionality. Traditionally, scientists have been good at sharing and reusing each other's application and infrastructure code. In order to take advantage of distributed computing resources in a grid, scientists will increasingly also have to reuse code, interface definitions, data schemas and the distributed computing middleware required to interact in a cluster or grid. The fundamental primitive that SOA infrastructures provide is the ability to locate and invoke a service across machine and organisational boundaries, both in a synchronous and an asynchronous manner. The implementation of a service can be achieved by wrapping legacy scientific application code and resource schedulers, which allows for a viable migration path. Computational scientists will be able to flexibly orchestrate these services into computational workflows. The standards available for service orchestration [7] and their implementation in industry strength products support the rapid definition and execution of scientific workflows [8].

An area that has so far being largely overlooked is that of providing appropriate programming language abstractions for science. Fortran and Message Passing Interface (MPI) are no longer appropriate in the setting described above. With the advent of abstract machines, it is now possible to mix compilation and interpretation as well as integrate code written in different languages seamlessly into an application or service. These platforms provide a sound basis for experimenting with and implementing domain-specific programming languages and we expect specialist languages for computational science to emerge that offer asynchronous and parallel programming models while retaining the ability to interface with legacy Fortran, C and C++ code.

Wolfgang Emmerich, M. Andy Parker, José Blakeley, Clemens Szyperski, Jamie Shiers, Vassily Lyutsarev

Semantics of Data

A revolution is taking place in the scientific method. "Hypothesize, design and run experiment, analyze results" is being replaced by "hypothesize, look up answer in data base" [9]. Databases are an essential part of the infrastructure of science. They may contain raw data, the results of computational analyses or simulations, or the product of annotation and organisation of data. Also, the current trend towards general access to knowledge in science is accelerating the worldwide publication of data. The development of an infrastructure for scientific data management is therefore essential. This poses major challenges for both database and programming language research, which differ from the conventional (business) requirements of databases. We attempt to describe some of them here.

A major issue is the *distribution* of data. Database technology has recognised for a long time that it is expensive or impossible to move large quantities of data. Instead one moves the code (software executing a program) to the data, and this is the core of distributed query optimisation. However, in distributed query optimisation, one traditionally thinks in terms of a small number of databases, but how do we optimise queries on, say, a sensor network in which each of a million sensors holds its own database? Second, we need to extend distributed query optimisation, which works for the simple operations of relational algebra, to work for more general operations that support scientific programming and to include, for example, spatial queries, string searches, etc. Known database techniques, such as parallel processing, set-oriented data access and intelligent indexing need to be extended, where possible, to support scientific data types. Third, we are facing much greater heterogeneity: individual data or document pieces require specific remote evaluation.

This distributed infrastructure will have to support stream processing and advanced data mining/machine learning techniques (see the section 'Prediction Machines'). We expect novel data mining methodologies and novel analysis techniques to be promising approaches to cope with growing data, especially where mathematical approaches have failed to yield a satisfying model to explain phenomena and where 'traditional' machine learning techniques have failed to bring back the knowledge out of the data. In the long run, an 'active learning' model is envisioned which requests data sources, like experiments, autonomously and leads to 'autonomous experimentation' (as described in the subsection 'Artificial Scientists' in Part 2 of this report).

But this is just the base technology that has to be developed. It must be supported by a computing environment in which it is easy for scientists to exploit the infrastructure. First and foremost is the *semantics* of data. This involves an understanding of the metadata, the quality of the data, where and how it was produced, intellectual property, etc. This 'data about data' is not simply for human consumption, it is primarily used by tools that perform data integration and exploit web services that, for instance, transform the data or compute new derived data. Furthermore, the environment should facilitate standard tasks such as querying, programming, mining or task orchestration (workflow) and it should

make it possible for scientists to generate their own computing tasks, rather than being reliant on database experts.

We believe that attempts to solve the issues of scientific data management by building large, centralised, archival repositories are both dangerous and unworkable. They are dangerous because the construction of a data collection or the survival of one's data is at the mercy of a specific administrative or financial structure; unworkable because of scale, and also because scientists naturally favour autonomy and wish to keep control over their information. When it is necessary to bring large quantities of data together for centralised computing, this should be done by replication, appropriate restructuring and semantic integration when necessary.

With this move towards reliance on highly distributed and highly derived data, there is a largely unsolved problem of preserving the scientific record. There are frequent complaints that by placing data on the web (as opposed to conventional publications or centralised database approaches), essential information has been lost. How do we record the details of the highly complex process by which a data set was derived? How do we preserve the history of a data set that changes all the time? How do we find the origin of data that has been repeatedly copied between data sources? Such issues have to be resolved to offer a convincing infrastructure for scientific data management.

Finally, we note that the future of databases in science is as much a social as a technical issue. Scientific funding organisations are increasingly requiring researchers to publish their data. But it is important that there are agreed community standards for publishing metadata, citations and provenance. Only if we have these will the data we are generating today be usable by applications of the future.

Peter Buneman, Serge Abiteboul, Alex Szalay, Andre Hagehülsmann

Intelligent Interaction and Information Discovery

A significant change in scientists' ability to analyse data to obtain a better understanding of natural phenomena will be enabled by (i) new ways to manage massive amounts of data from observations and scientific simulations, (ii) integration of powerful analysis tools directly into the database, (iii) improved forms of scientist-computer-data interaction that support visualisation and interactivity, (iv) active data, notification, and workflows to enhance the multi stage data analysis among scientists distributed around the globe, and (v) transformation of scientific communication and publishing.

Managing the Data Explosion

It should be abundantly clear from this report that the amount and complexity of scientific data are increasing exponentially. Scientists have difficulty in keeping up with this 'data deluge' [10]. It is increasingly clear that, as a consequence, the way scientists interact with the data and with one another is undergoing a fundamental paradigm shift. The traditional sequence of 'experiment › analysis › publication' is changing to 'experiment › data organisation › analysis › publication' as more and more scientific data are ingested directly into databases, even before the data are analysed (see also section 'Transforming Scientific Communication').

Today, data are not only generated by experiments, but by large numerical simulations. The size of these simulations is such that there is as great a challenge in storing and retrieving the results for subsequent analyses as there is in performing the computations themselves. The challenge is to extract information and insights from the data without being hindered by the task of managing it. How can scientists interact with their data in such a world?

Adaptive organisation and placement of data and computation

Since network speeds to most academic locations are not keeping up with the size of and demand for data, in many cases scientists will not be able to copy data to their own machines; the analysis needs to be run closer to the data. As a result, data archives will have to offer access to analysis tools (and computational resources) and provide some 'private' workspace – all this will allow for laboratory – and discipline-spanning collaboration while also helping to curb the exploding network traffic. In other cases, repeated use of data sources, need for specialised software, or latency concerns would dictate that data be moved closer to the computation. Also, groups of scientists need to carry out their analysis tasks on well-defined, coherent data subsets. For these reasons, intelligent, robust, dynamic algorithms are needed for determining and continuously re-evaluating the best placement of data replicas and computation in a large-scale, heterogeneous computing environment. Data stores will need to be capable of being extended to absorb the software packages containing the algorithms for data analysis required by scientists, better divide-and-conquer techniques are needed to help break through the polynomial complexity of existing algorithms, and better distributed, and loosely-coupled techniques (e.g. Web services) are required in order to distribute, exchange, and share results among expert scientific communities.

Most scientists will only look at a small part of the available data. If this 'hot' data is mirrored at several locations, and this hierarchical process is repeated at several levels, one can have a system where both the I/O and the computational load are much better distributed. As a result, large databases will be complemented by a federated hierarchy of smaller, specialised databases. This is the approach taken by the particle physics community dealing with data from the Large Hadron Collider, where they organise the data to reside in a hierarchical multi-tiered system [11]. Similar approaches are also well established in the commercial realm through the use of specialised 'Datamarts' that sit in front of the larger and more complex Data Warehouses of large organisations.

There are also challenges within the scope of an individual processing cluster. Many Beowulf clusters built over the last decade are I/O poor. In order to be able to perform such data intensive computations successfully, we will also need balanced systems [12], where there is adequate I/O bandwidth to deliver the data to the CPUs. A further concern with existing cluster systems is that their file

systems tend to be optimised for raw throughput rather than for interaction. This limits the performance that can be obtained by data-centric workflow systems that are necessary to move from a batch-oriented approach to a more interactive one, in which scientists can control the processing based on visualisations and real-time analysis.

Tools for data analysis

The demand for tools and computational resources to perform scientific data analysis is rising even faster than data volumes. This is a consequence of three phenomena: (i) more sophisticated algorithms consume more instructions to analyse each byte; (ii) many analysis algorithms are polynomial, often needing N^2 or N^3 time to process N data points; and (iii) I/O bandwidth has not kept pace with storage capacity. In the last decade, while capacity has grown more than 100-fold, storage bandwidth has improved only about 10-fold.

These three trends, algorithmic intensity, non-linearity, and bandwidth limits mean that the analysis is taking longer and longer. To ameliorate these problems, scientists will need better analysis algorithms that can handle extremely large datasets with approximate algorithms (ones with near-linear execution time), they will need parallel algorithms that can apply many processors and many disks to the problem to meet CPU-density and bandwidth-density demands, and they will need the ability to 'steer' long-running computations in order to prioritise the production of data that is more likely to be of interest.

Integrated symbolic computation, data mining and analysis

After seeing a pattern in a scientific data set, the next step is to explain it. Scientists use packages such as Maple™, Mathematica® and MATLAB® to aid in lightweight numerical analysis, prototyping and hypothesis formation. Bringing symbolic computation tools closer to the database and to the mainstream deployment programming languages in integrated development environments, and enabling symbolic code and prototype mathematical models to be translated directly into deployable code with database query and visualisation just a click away will enhance scientists' analysis significantly.

Data mining algorithms allow scientists to automatically extract valid, authentic and actionable patterns, trends and knowledge from large data sets. Data mining algorithms such as automatic decision tree classifiers, data clusters, Bayesian predictions, association discovery, sequence clustering, time series, neural networks, logistic regression, and linear regression integrated directly in database engines will increase the scientist's ability to discover interesting patterns in their observations and experiments.

Type systems for units, precision, uncertainty and error propagation

The 1999 infamous crash of the Mars Climate Observatory due to a mismatch of metric and imperial measurement units spurred renewed interest in programming language technology to head off a repeat. It is increasingly compelling to integrate precision and accuracy in type systems, and to develop first-class data types that perform commonplace scientific error propagation. For instance, the type of a measurement of force in pounds ought to include, perhaps, its one-sigma uncertainty. Extending database query, search, and data mining engines to incorporate units, precision, uncertainty, and error propagation as an integral part of expression evaluation services will bring new levels of accuracy to the scientist's analysis toolkit.

Data cubes, data visualisation and rapid application development

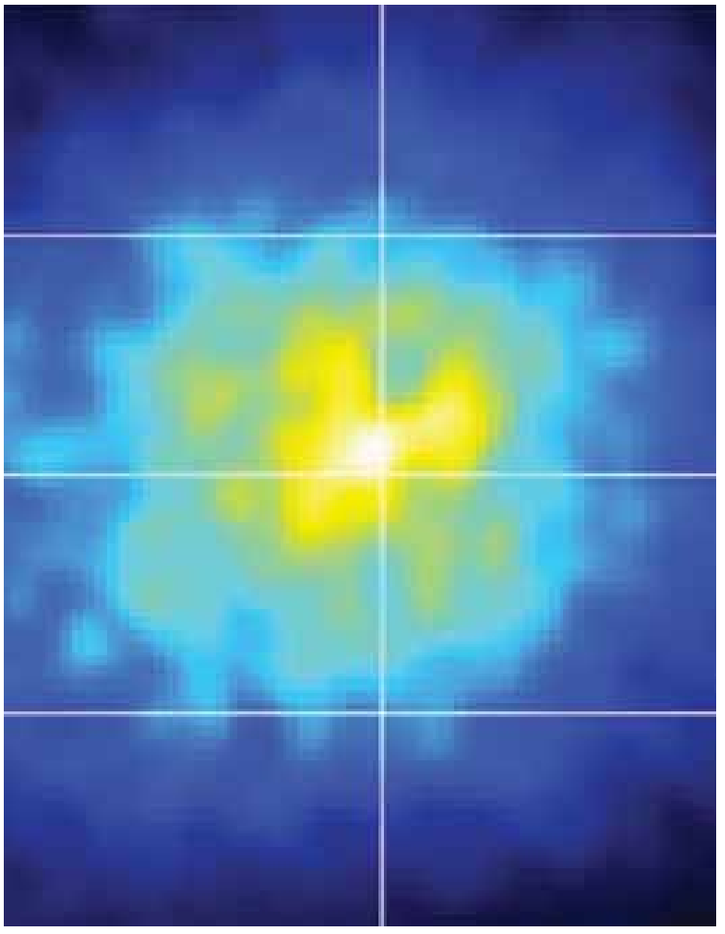
Large observational data sets, the results of massive numerical computations, and high-dimensional theoretical work all share one need: visualisation. Observational data sets such as astronomical surveys, seismic sensor output, tectonic drift data, ephemeris data, protein shapes, and so on, are infeasible to comprehend without exploiting the human visual system. For instance, cosmic filaments would never have been found without the visualisations of the Harvard-Smithsonian Center for Astrophysics catalogue. Similarly, finite-element simulations, thunderstorm simulations, solid-state physics, many-body problems, and many others depend on visualisation for interpretation of results and feedback into hypothesis formation. Finally, some frontiers of pure theory, especially where large numbers of dimensions are involved, are exploiting visualisation to aid intuition and communication of results.

Many scientists, when faced with large amounts of data want to create multi-dimensional aggregations, where they can experiment with various correlations between the measured and derived quantities. Much of this work today is done through files, using home-brew codes or simple spreadsheets. Most scientists are not even aware that tools like Online Analytical Processing (OLAP) data cubes are available as add-ons to the database engines. Smart data cubes play a twofold role. First, they serve as caches or replicas of pre-computed, multi-dimensional aggregations that facilitate data analysis from multiple perspectives. Second, they support the visualisation of data over data partitions. Given the deluge of data scientists need to deal with, we also need to use data mining techniques to facilitate automatic detection of interesting patterns in the data.

An important way for database technology to aid the process is first through transformation of schematised large-scale science data into schematised small-scale formats, then through transformation of small-scale formats into standardised graphical data structures such as meshes, textures and voxels. The first kind of transformation fits into the category of OLAP, which is a staple of the business community. The second kind of transformation is an exciting area for applied R&D.

Empowering data-intensive scientists

The final piece that brings all the above advances in data management, analysis, knowledge discovery and visualisation together to empower the scientist to achieve new scientific breakthroughs is a truly smart lab notebook. Such a device would unlock access to data and would make it extremely easy to capture, organise, analyse, discover, visualise and publish new phenomena [13]. While several electronic lab notebooks are already on the market, none fulfil the requirements



Galactic centre, gamma ray image

It is thought that the annihilation of dark matter particles and antiparticles forms gamma rays that have a certain energy (511 – keV). The gamma ray intensity is greatest (yellow) at the centre of the Milky Way. The plane of the galaxy is horizontal. Dark matter is the non-visible matter thought to make up most of the mass of the universe. The Milky Way's centre was observed by Integral, the ESA's gamma ray space telescope, and modelled by researchers Dr Celine Boehm and Dr Dan Hooper. The model uses low-mass (relative to previous theories) particles of dark matter. Results published in 2003.

James King-Holmes / SCIENCE PHOTO LIBRARY

of scientists well, nor the criteria for the functionality of such a system outlined here. However, the outline of developments under way presented here suggests that a truly smart lab notebook will be in scientists' hands quite some time before 2020.

Summary

The challenges of modern science require an intense interaction of the scientists with huge and complex data sets. The globally distributed nature of science means that both scientific collaborations and the data are also spread globally. As our analyses are becoming more elaborate, we need advanced techniques to manipulate, visualise and interpret our data. We expect that paradigm will soon emerge for the scientist–data interaction which will act as a window into the large space of specialised data sources and analysis services, making use of all the services mentioned above (discovery of data and analysis services, data administration and management tasks) in a way that is largely hidden to the scientist. Many sciences share these data management, analysis and visualisation challenges, thus we expect a generic solution is not only possible but will have a broad impact.

Alex Szalay, José Blakeley, Michael Franklin, Brian Beckman

Transforming Scientific Communication

The Web and associated technical advances will dramatically shape scientific publishing and communication over the next 14 years. These changes will occur in five main areas of development: (i) interactive figures and new navigation interfaces; (ii) customisation and personalisation; (iii) the relationship between journals and databases; (iv) user participation; (v) searching and alerting services.

Perhaps the greatest effect of the Web on science to date has been seen in scientific publishing or, more broadly defined, in scientific communication. Given that science is a global endeavour and that the web is arguably the most effective global communication medium yet devised, this should not come as a surprise. Yet the potential for the web to completely reshape scientific communication - and in doing so to reshape scientific research itself - is underestimated.

The effects of the Web on scientific publishing as of 2005 have focused heavily on the issue of *open access*. At its core, this is a debate about whether it is more effective for the publication of scientific papers to be paid for by authors or by readers (and their respective institutions or funding agencies). We believe this debate is almost insignificant compared to the changes that the Web will eventually have on scientific communication. Those who focus on open access, far from being radical, are not being nearly radical enough.

The grand challenge for scientific communication is not merely to adjust the economics of publishing to reflect new realities (though that is certainly happening), but rather to redefine the very concept of a scientific publication. Only in this way will scientific publishing remain relevant and fulfil its duty to help accelerate the pace of scientific discovery now that we are unconstrained by many of the restrictions imposed by print.

The changes afoot can be usefully considered in five areas: Data display, Dynamic delivery, Deep data, Discussion and dialogue, and Digital discovery.

Data display

One of the most obvious ways in which online scientific publications can improve is to provide the reader with a degree of interactivity, especially in figures. Currently, such functionality in scientific publications is somewhere between very rare and non-existent. The most obvious ones are the visualisation formats such as Flash® and Scalable Vector Graphics (SVG). Furthermore, applications of Flash®, SVG and similar technologies are not limited to figures. For example, they should also prove useful in providing new search and navigation interfaces. Within the next 10 years, we expect to see the development of a radically different yet effective navigation system.

Dynamic delivery

Online pages can be generated the moment they are requested, thus allowing customisation (according to a particular time or place) and personalisation (according to a particular user). Personalisation on the Web has had a long – and not always particularly happy – history. But the latest offerings from companies such as My MSN®, My Yahoo!® and Google™ News and My Google™ show promise. They should become even more compelling and pervasive as they grow in their ability to infer users' interests from their behaviour instead of requiring users to define their interests explicitly.

Scientific content, too, is ripe for personalisation. First and most straightforwardly, different types of readers are looking for very different things when they read the same scientific paper. Some, reading outside their main area of study, may only want a brief, superficial summary. Others may want only to scan the abstract and figures. And others still may want to read the whole paper, including accompanying supplementary information and detailed experimental protocols. To try to serve this range of interests with one document is next to impossible. But in the online world it is much easier to provide different readers with different lengths and depths of content depending on their areas of interest and expertise. Indeed as the online version of a research paper comes to be seen as primary, it is likely that within a 10-year timeframe, the print versions of at least some journals will stop including the full text of papers and will carry instead only summaries and commentaries with pointers to the full content online.

Deep Data

Modern scientific communication is dominated by journals and databases, which is quite appropriate in the sense that each serves rather different, and in many ways complementary, purposes. Even so, it is disappointing both that they are so poorly integrated with one another, and that each has not adopted more of the strengths of the other. However, within 5 years, we should see much richer mutual linking between journals and databases, and in a 10 or 15 year timeframe we will see the rise of new kinds of publications that that offer the best of both of these worlds. However, linking in a reliable, scientifically meaningful way is difficult - so difficult that it requires significant effort by an editor or another domain expert. Nevertheless, we expect this problem to be significantly overcome across much science only in a timeframe of 10-15 years.

Yet, far from limiting themselves to merely *linking* to databases, scientific journals will in some senses need to *become* databases. Initially this will manifest itself in the way that papers handle accompanying data sets. In the longer term, though, hybrid publications will emerge that combine the strengths of traditional journals with those of databases. We are likely see a new breed of scientific publication emerge on a timescale of about 10 years that will cater primarily for researchers who wish to publish valuable scientific data for others to analyse. The data will be peer-reviewed and the author will get credit for having published a paper even if the information contained does not explicitly present any new scientific insights. The main technical challenge here is the sheer volume of data. Though the difficulties may be alleviated somewhat by massively distributed data storage and sharing networks, we expect this problem to still be with us 14 years from now.

Just as crucial as being able to give data sets a suitable status within a paper is the ability of publishers to accept and publish data sets in *structured* and *machine-readable* formats. Indeed, publishers also have a role in helping to promote the use of such formats. To give one example, *Molecular Systems Biology* [a journal launched jointly by Nature Publishing Group (NPG) and the European Molecular Biology Organization (EMBO)] encourages authors of papers describing computational models of molecular pathways to submit their models using Systems Biology Markup Language (SBML; http://www.sbml.org/).

Discussion and dialogue

Away from scientific publishing, the meme of the moment is the 'two-way web' in which users are not merely passive consumers but active participants. This is perhaps most evocatively expressed in the term 'architectures of participation', a phrase popularised by technical book publisher and Web guru, Tim O'Reilly, originally referring to open-source software projects, but since then also a common way to describe the way in which certain websites (e.g. eBay®, Blogger™, and Wikipedia) create environments in which users contribute content and services, and generally interact with each other, without directly involving the service provider. Another example is social bookmarking services such as Connotea, which caters specifically for the needs of scientists (http://www.connotea.org/). It seems clear that services like these will become an important way for scientists to organise, share and discover information, building and extending on-line collaborative social networks.

Digital Discovery

As the volumes of scientific text and data continue to balloon, finding timely, relevant information is an increasing challenge for researchers in every discipline. Scholarly search services such as PubMed, Google™ Scholar and Astrophysics Data System certainly help a lot. And, although most scientists are

unaware of them, so do 'content locator' technologies such as OpenURL (http://www.exlibrisgroup.com/sfx_openurl.htm) and DOIs (or Digital Object Identifiers, a general metadata system currently most widely used to enable cross-publisher linking of citations to papers). It is not practical to attempt to capture everything a paper contains – present-day ontologies and data models are nowhere near as expressive as human languages – but in principle, we can provide a useful summary of the bibliographic details, authors, institutions, methods and citations, as well as the main scientific entities (molecules, genes, species and so on) with which the paper is concerned. This, in turn, should enable much more specific searching of, and linking to, the paper in question. With appropriate metadata, it would even be possible to conduct searches for 'papers that disagree with this one', a concept unimaginable with even the best search engines today. The main difficulty here is collecting the necessary information in a suitably structured form. We expect to see major progress in this area over the next 14 years.

The scientific paper as a means of communication is here to stay for the foreseeable future, despite the continuing online revolution. But it will inevitably evolve in response to scientific needs and new enabling technologies. As with the evolution of organisms, this will involve a large number of incremental changes that will collectively represent something of a revolution. New functionality will be provided in the online versions of papers and their relationships with their print versions will be redefined. We will also see the rise of new kinds of publications, not merely with different business models, but also with different editorial and technical approaches. This will create greater diversity among scientific publications as they strive to serve different research needs. And those needs will also evolve as science itself changes in response to further technical advances. This means that the scientific publishing and communications industry will need to continually adapt, and at a faster pace than in the past. These developments will not only reflect changes in the way research is done but in some cases may also stimulate them.

Timo Hannay

Computational Thinking

This report argues strongly that computer science can make a major, if not reforming contribution to the natural sciences. Natural sciences are defined with reference to the world in which we live as the subject and the scientific methods of empirical study and postulation of laws and theories to explain what is observed. Computer science as a discipline is harder to define: it does not have the empirical foundations of the natural sciences, it is more than just symbolic reasoning (i.e. mathematics) and it is not just a compendium of engineering principles and technology. For that reason, at this point in the document, we set out in broad terms what we believe computer science is so as to anchor the subsequent discussion.

Computer science is perhaps best characterised by the way in which computer scientists approach solving problems, designing systems and understanding human behaviour in the context of those systems[2]. Within computer science, there is a strong body of theory that explains the potential and limits of computation, what we might call 'computational thinking', a term coined by Professor Jeanette Wing, head of Computer Science at Carnegie Mellon University, Pittsburgh, USA. She defines computational thinking on her web page [14] from which the following is extracted:

> *Here is my grand vision for the field: Computational thinking will be a fundamental skill used by everyone in the world by the middle of the 21st Century. To reading, writing, and arithmetic, add computational thinking to every child's analytical ability. Imagine! And just as the printing press facilitated the spread of the 3 R's, what is deliciously incestuous about this vision is that computing and computers will facilitate the spread of computational thinking. What do I mean by computational thinking? It includes a range of "mental tools" that reflect the breadth of our field. When faced with a problem to solve, we might first ask "How difficult would it be to solve?" and second, "What's the best way to solve it?" Our field [computer science] has solid theoretical underpinnings to answer these and other related questions precisely. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation. Computational thinking is type checking, as the generalization of dimensional analysis. Computational thinking is choosing an appropriate representation for a problem or modelling the relevant aspects of a problem to make it tractable. Computational thinking is using abstraction and decomposition when tackling a large complex task or designing a large complex system. It is having the confidence that we can safely use, modify, and influence a large complex system without understanding every detail of it. It is modularizing something in anticipation of multiple users or pre-fetching and caching in anticipation of future use. It is judging a system's design for its simplicity and elegance. It is thinking recursively. It is thinking in terms of prevention, protection, and recovery from worst-case scenarios (violated pre-conditions, unpredictable environments) through redundancy, damage containment, and error correction. It is calling gridlock deadlock and learning to avoid race conditions when synchronizing meetings. Computational thinking is even using the difficulty of solving hard AI [computational] problems to foil computing agents, e.g. as CAPTCHAs are used daily by websites for authenticating human users. [A CAPTCHA is a program that can generate and grade tests that most humans can pass but current computer programs can't, for example recognize words displayed as distorted text.] In short, computational thinking is taking an approach to solving problems, designing systems, and understanding human behaviour that draws on the concepts fundamental to computer science.*
>
> *Andrew Herbert*

[2] This characterisation is the work of Professor Jeanette Wing of Carnegie Mellon University, and presented at the Microsoft Research Asia 'Computing in the 21st Century' Conferences in Hangzhou, China and Hong Kong, November 2005.

2 Building Blocks of a Scientific Revolution

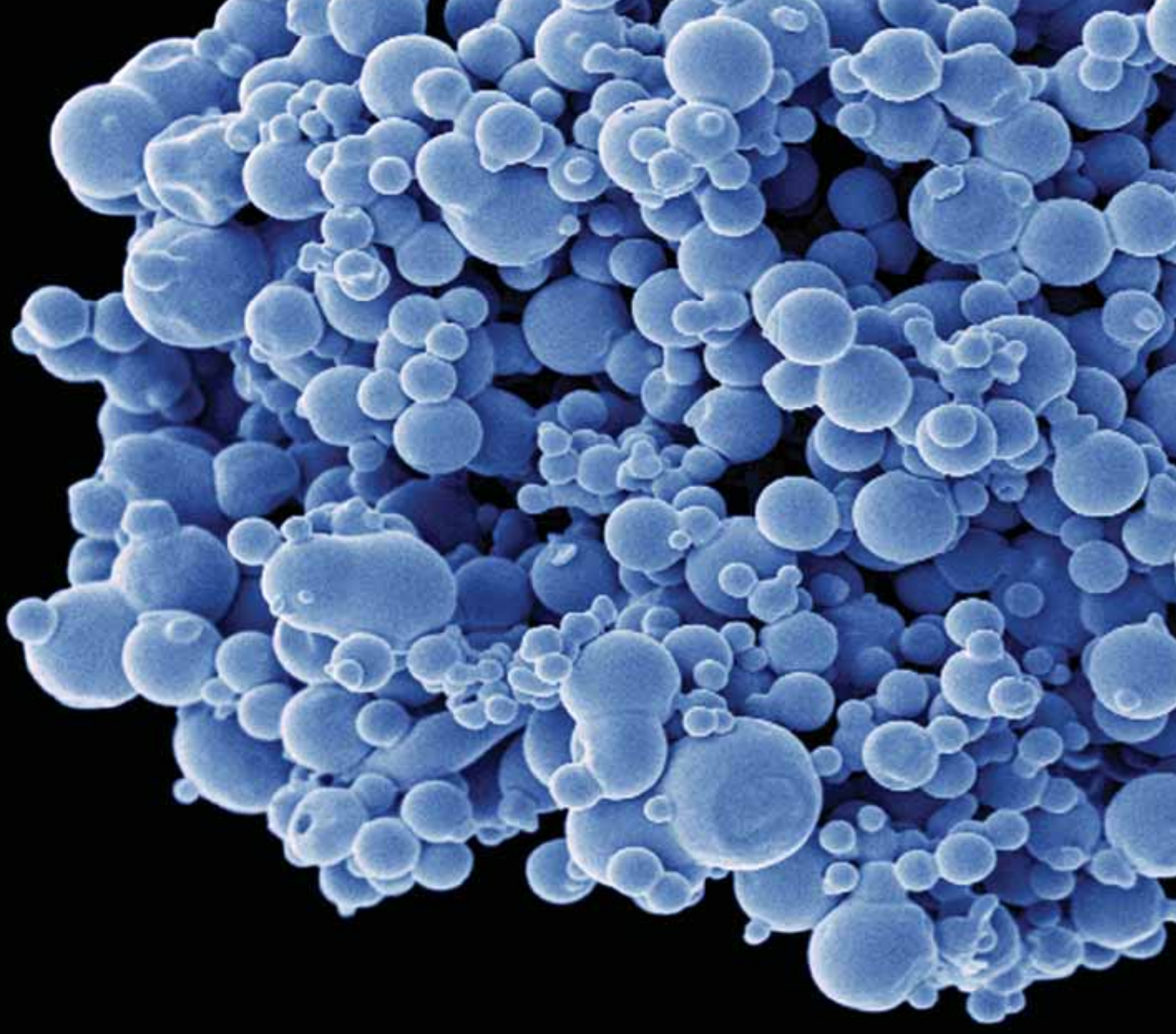
Concepts, Theorems and Tools developed within computer science are now being developed into new conceptual tools and technological tools of potentially profound importance, with wide-ranging applications outside the subject in which they originated, especially in sciences investigating complex systems, most notably in biology and chemistry.

We believe these tools have the potential to have a fundamentally radical impact in science, and especially in the biological sciences. In Part 2 we explain why, and introduce some of these tools. We believe such tools will become integrated into the fabric of science, and are the potential starting point for fundamental new developments in biology, biotechnology and medicine, as well as other branches of science towards 2020, and discussed in Part 3.

Liposome vesicles

Coloured scanning electron micrograph (SEM) of liposome vesicles. These artificially-constructed, spherical vesicles possess a selectively-permeable membrane that closely resembles the membrane of a living cell. They are used in biotechnology research to investigate the functioning of the cell membrane and, since they can be incorporated into living cells, are used to deliver drugs of high toxicity to specific cells in the body, such as cancer cells. They are also used in cosmetics.

David McCarthy / SCIENCE PHOTO LIBRARY



The fundamental role of computer science concepts in science

Part 1 outlined how computers will play an increasingly important and eventually ubiquitous role in most branches of science, and how they are changing how scientists work. Altogether more radical, however, is the importance of *computer science*. We believe that computer science is poised to become as fundamental to science, and in particular the natural sciences, as mathematics has become to science, and in particular the physical sciences.

Two important pillars underpin this statement: First, computer science concepts and theorems deal with dynamics in a discrete and reactive sense. Calculus, for example, and its more modern derivatives (excuse the pun) is the main way in which mathematics deals with dynamic issues, but it does so in a continuous fashion, with continuous kinds of cause-and-effect; it deals with rates of increase, with feedback loops, with growth and movement, etc. In contrast, computer science deals predominantly with the interactively discrete, which is really what is meant by the term *reactive*, and it is also able to combine this with the continuous. In fact, computer science is the science *dedicated* to the dynamic. In most kinds of complex systems, biology perhaps being the primary example, the discrete is not only more central but is also much harder to deal with. Indeed, biological systems are the most exciting dynamic systems we will ever know; they are predominantly reactive, and they not only behave but also affect, prescribe, cause, program and blueprint other behaviour. In short, the characteristics of computer science are central to the dynamics of biological systems: concurrency, time dependence, cause-effect phenomenon and distributed control.

Second, computer science is also about algorithms and programs, that is, with *generic prescriptions* for creating dynamics. It not only analyses dynamics and writes equations that capture dynamic phenomena, which is what the dynamic parts of mathematics do well (for the continuous case), but computer science *builds* dynamics. And it is this, perhaps more than anything else, that gives computer science some of its most important and special ways of thinking, its tradition and its nature.

Given that many of the most important and fundamental challenges and opportunities for the 21st Century can be characterised by their complexity and dynamics, then computer science clearly – we claim and make a case for here – will be equally fundamental to addressing them. Part 3 of this report outlines some examples of how.

One of the first glimpses of the potential of computer science concepts and tools, augmented with computing, has already been demonstrated in the Human Genome Project, and by the success of structural biology to routinely decipher the three-dimensional structure of proteins. In this and in related sequencing projects, scientists use computers and computerised DNA sequence databases to share, compare, criticise and correct scientific knowledge, thus converging on a consensus sequence quickly and efficiently [15]. These branches of biology succeeded in unleashing the power of computers to their benefit because both have adopted good mathematical abstractions to describe their research such as: the 'DNA-as-string' abstraction (a mathematical string is a finite sequence of symbols) to describe DNA sequences, and the 'protein-as-three-dimensional-labelled-graph' abstraction, to describe the three-dimensional structure of proteins. Armed with good abstractions, these scientists were able to code their knowledge in a mathematical form that is amenable to processing and sharing via computers. We expect that the rest of biology and other scientific disciplines will also be able to make such big strides, with the aid of computers and computer science concepts and tools, by adopting similarly useful abstractions for more complex systems and processes, as explained in the subsection 'Codification of Biology' in the section 'New Conceptual and Technological Tools' below.

The coding of scientific knowledge will not only empower scientists by allowing them to share, compare, criticise and correct scientific knowledge via computers, it will also enable a change in the way science is done. Coded scientific knowledge can be analysed computationally, before any experimentation. It can be checked, computationally, for consistency among coded theories, and for consistency between theories and accumulated data, akin to computer program debugging [16]. When inconsistency among theories is uncovered, it might be resolved by computer-designed 'crucial experiments' [17-19]. Furthermore, computational analysis of theory versus experimental data may suggest additional experiments to be performed, manually or automatically, as described later in the sections 'Integrating Theory, Experiments and Models', and the section 'New Conceptual and Technological Tools, in the subsections 'Artificial Scientists' and 'Prediction Machines'.

We believe that the concepts and tools developed in computer science over the past 70 years will be useful not only at the *'meta level'*, in helping to manage and develop theory, data and experimentation, but most importantly, also at the *'object level'*, in helping to form scientific theories. For example, computer systems and biomolecular systems both start from a small set of elementary components from which, layer by layer, more complex entities are constructed with ever-more sophisticated functions. Computers are networked to perform larger and larger computations; cells form multi-cellular organisms. All existing computers have an essentially similar core design and basic functions, but address a wide range of tasks. Similarly, all cells have a similar core design, yet can survive in radically different environments or fulfil widely differing functions. Hence we believe the abstractions, tools and methods used to specify and study computer systems should illuminate our accumulated knowledge about biomolecular systems [15].

Several fundamental computer science concepts are already on their way to becoming household names in science, and many more will follow. For example, abstraction is a fundamental tool in computer system design: when designing a complex computer system, identifying the right levels of abstraction within the system is perhaps the single most important design decision. Within a computer system, one can easily find a dozen or so such levels, starting from logic gates, logic circuits, functional units, hardware devices, microinstructions, abstract machine

and the machine language, abstractions for memory and communication, high level language, procedures, data types, algorithms, system design, and system specification. Analogously, identifying levels of organisation in biological systems was fundamental to progress in biology: biomolecules (DNA, RNA, proteins), biomolecular machines (polymerases, ribosome, spliceosome) and functional molecular complexes (membranes and pores), signalling pathways, organelles, cells, organs, organisms, and beyond.

As another example, the concepts developed in algebraic concurrency theory, such as concurrency, indeterminism, communication, synchronisation, processes, channels, and messages, may prove essential for the full understanding and codification of complex inter- and intra-cellular biological processes [20]. As a third example, we expect core computer science concepts on interchangeability of program and data, universal computers, interpreters, compilers, meta-interpreters, partial evaluation, and compositional semantics, to prove essential for the full understanding of the role of DNA as program and data, of the universality of cellular design, and of gene regulation and specialisation. As a fourth example, consider the complexity of each biological unit, and organism as a whole, as encoded in its genome. We expect the notion of descriptive complexity, developed by Kolmogorov [21], to play an essential role in understanding and measuring biological complexity at all levels. As a fifth example, modularity and well-defined interfaces are key attributes of good computer design. They ensure that errors in one component may have a limited effect on other components, and therefore can be tracked and corrected. They also ensure that the design can easily be changed and evolved as requirements change. Similarly, we believe that modularity became a fundamental attribute of the evolvable components of biological systems, as non-modular designs were not able to evolve and survive through changes in external conditions. Uncovering the modularity and interfaces of evolvable biological systems is a major challenge of biology, and a computer science perspective on these issues might be of assistance. In general, such advances in science will rely on the development and application of new conceptual and technological tools, discussed in a later chapter.

Ehud Shapiro, David Harel, Christopher Bishop, Stephen Muggleton

Integrating Theory, Experiments & Models

The integration of theory, experiments and models is a central, and challenging, goal in science. Achieving this goal fully would dramatically increase our understanding of natural phenomena and enable revolutionary advances in science. It is also a goal that will be increasingly challenging as computing enables the construction of ever more complex models and experiments, and produces data of increasing complexity and volume. As we shall see later, achieving this goal by 2020 is not only necessary in areas such as understanding earth systems and biological processing, but also looks increasingly possible in several branches of science through new kinds of conceptual and technological tools provided by computer science. Achieving this goal also involves marrying computer science, computing and the scientist.

Articulation of models and experiments

The accumulation of large-scale data in science – whether the result of high throughput techniques in genomics, proteomics, or metabolomics, or combinatorial chemistry, astronomy, high-energy physics or earth sciences – and a move to the forefront of large-scale computational modelling are already making significant demands on computing beyond the current state-of-the-art. In the case of large-scale data, as previous sections in this report have outlined in some detail, it must be stored and managed alongside appropriate metadata so that its meaning and provenance can be established, and retrieval must be rapid and transparent with respect to data distribution, irrespective of the nature of the data. Large-scale computational models must be constructed from components, managed and exchanged between modellers, and executed or analysed across heterogeneous tools and computational platforms. Whilst none of the above is surprising, what is surprising is that science largely looks at data and models separately, and as a result we miss the principal challenge – the articulation of modelling and experimentation. Put simply, models both consume experimental data, in the form of the context or parameters with which they are supplied, and yield data in the form of the interpretations that are the products of analysis or execution. Models themselves embed assumptions about phenomena that are the subject of experimentation. The effectiveness of modelling as a future scientific tool and the value of data as a scientific resource are tied into precisely how modelling and experimentation will be brought together.

The classic picture of how this is done is as follows: a model is constructed as a 'theory', a set of inputs are provided to the model and when the model is analysed or executed, a set of behaviours are observed. These behaviours are compared with those of the domain under a similar set of conditions and if the correspondence between behaviour of the model and the domain holds over some range of inputs, this tends to lend weight to the theory. Once sufficient confidence is established in the model it can be used in place of experiment in the context of, or as the input to, other models. This is, of course, highly simplistic.

The correspondence challenge

In a real setting, there are no clean sets of inputs to be supplied to a model. Experimental data are contested, the methods by which data are obtained may give rise to inaccuracies or noise, and most confusingly the model may itself be tainted with assumptions. Establishing the correspondence between model behaviour and domain behaviour can be very difficult. It is unlikely, given the simplifications entailed in model building, that the model will behave precisely as the real-world system does. For example, the behaviour of the model may be congruent with that of the domain within a limited range of inputs – much like Newtonian physics holds within a wide range but ultimately yields to quantum mechanics at one end and to relativity at the other. Comparing the idealised behaviour of the model against that of the real-world system poses questions of

what is essence and what is accident. The behaviour of the model may, of course, correspond to that of the domain but may be the result of a different mechanism. More straightforwardly, the model may be buggy in the sense of not being a secure implementation of the theory it is intended to represent. This much more complex and nuanced role for models suggests an equivalently complex embedding of modelling within the experimental setting.

Though experimental methodology is mature, and a highly developed scientific culture has built up around it, this is not true of modelling. Computational modelling in the sciences, however mathematically sophisticated, is currently methodologically the domain of 'hackers'. The intertwining of experimentation and modelling requires the methodologies to mesh as well as the data and computational resources. This is difficult to achieve: practically, the time frames of experimentation and modelling are very different and resourcing and mindsets are also different. Particularly tricky is that the sort of data required by modellers is not always preserved through to publication in 'convergent' hypothesis-driven research, suggesting another way in which model and experiment may interrelate as modelling shapes the data gathering agenda.

What are the implications of this discussion? We believe it suggests an important agenda, one that has previously been neglected. Standing between large-scale, static, data curation and computationally demanding models, we require a sophisticated framework within which data, models and their complex dynamic relationships with each other can be managed. There are, in particular, compound versions and configurational relationships which must be maintained. Scientific rationale is essential for understanding the nature and status of the relationships. It is far from clear that existing technologies are capable of supporting this, strained as they are by the more straightforward tasks of scientific data management. It will require novel data models, architectures and highly sophisticated version control and support for fine-grain scientific workflows of a very different kind to be integrated into the very fabric of the scientific process.

Anthony Finkelstein

From Complexity to Coherence

Many of the most pressing and important aspects of life, such as biology and medicine (e.g. intracellular networks, organ systems, epidemiology), the environment (e.g. ecosystems, earth systems interactions), social systems (e.g. transport, cities, relationships), communication networks and commerce are represented as complex systems. The greatest challenges in science are to understand these *complex systems*.

But whilst 'complexity' is often seen as simple elements interacting to produce 'complex behaviour', it is often the reverse – highly complex elements producing coherent behaviour. Perhaps the most important scientific challenge of all is to be able to understand and predict how such complex systems produce *coherent* behaviour.

Complexity has come to be seen as representing a scientific frontier, alongside the frontiers of the very small, the very large, the very high energy and so on, and an increasing ability to interact systematically with systems of high complexity will have very profound effects in future science, engineering and industry, and in the management of our planet's resources.

Three prominent branches of science address themselves directly toward understanding highly complex systems: biology, computer science, and complexity theory. There are opportunities to cross-fertilise the intellectual heritage of formal methods and ideas in all three disciplines, and to reconcile different definitions of, and approaches to, complexity, which currently are not well integrated. Even in computer science, several types of complexity are measured and studied. Time – and – space complexity of algorithms may be the most prominent, where one seeks a function that takes as input the size of the problem, and produces as output the time (or space) required by the algorithm to solve it. Another type of complexity, which may be perhaps the most relevant to biology, is 'descriptive' complexity, also called Kolmogorov complexity. The Kolmogorov complexity of a string is the shortest program that can produce that string. Kolmogorov complexity is relevant to the study of genomes and other biological descriptions, such as, for example, lineage trees.

Complexity in natural and social sciences typically arises in systems with large numbers of components and a large number of structured interactions between them. Large numbers of components and large numbers of interactions are not themselves diagnostic of a complex system (gases, for example, have both, but their behaviour is well predicted by statistical dynamics). Structured or patterned interactions in multi-component systems render accurate prediction of the system's behaviour very difficult indeed (see the subsection 'Codification of Biology' in the following section).

Complex systems, science and computer science

Understanding complex systems is aided by the 'fundamental theorem' of computing science: Shannon's measure of Information. A natural representation for complex systems of N components is as a matrix of interconnected elements, in which the structure of interactions is present in the pattern of connections in the matrix[3]. This has the consequences that the types of mathematical and computational tools that are appropriate to such systems change (e.g. from systems analysis to topological analysis); that the function or failure of individual components or interactions becomes much less important than the statistical dynamics of combinations of elements, as in 'network' or system failures; and that the failure modes of the system move from being visible to being invisible, without computational help. As N rises, the behaviour of a system becomes much less intuitive, until, for large N systems, it can be strikingly counterintuitive. Computational approaches that can render these counterintuitive features of these systems visible, and form the basis of predictive capability, are the key to managing the complex systems around us more successfully.

Topological structure in complex systems

A major problem that generally confounds understanding of complex and biological systems is the problem of determining structure and topology in the networks that define them. Understanding how a system is organised is an unavoidable prerequisite for understanding how it works. How can this structural information about very complex networks be derived? This challenge is being addressed systematically in relation to large-scale brain networks, networks of interactions within proteomes and metabolic networks, networks of trading relations between economic actors, and the Internet. These approaches will become more generally capable of determining structure in any very complex network, even very large ones. Sparsity, asymmetry, and visualisation become important problems for large systems, and will increasingly need to be actively researched.

Vulnerabilities in complex systems

Recent results have shown that most biologically important complex networks share elements of topology. Now-classical studies have shown that the connectivity in real networks (such as metabolic and protein–protein interaction networks) is described fairly well by a power-law distribution. It is a robust property of all these systems that their network integrity is degraded only very slowly by random failure or deletion of nodes, and that network integrity is strikingly vulnerable to intelligently targeted attack.

Work to date has focused very largely on the 'scale-free' topology and on 'hubs' as the network property of nodes that determines their importance to network integrity and so function. However, scale-freeness is only one aspect of the topology of most complex networks, and a number of network properties beyond hubness have also already been shown computationally to determine network integrity and function. These new computational insights are important for understanding, manipulating, repairing or destroying complex networks, and we envisage significantly greater theoretical and practical understanding of the 'control points' of complex systems to emerge from focused computational approaches.

In view of these ideas, one can ask what is needed to progress from complexity to coherence. Further progress will depend on improved abilities to measure

Computer model of a chemical wave

The model was produced by combining seven differential equations using the principles of chaos mathematics. Chemical waves are the result of dynamic processes in Belousov-Zhabotinsky (BZ) reagents. This model shows a remarkable agreement with observed chemical wave structure. Such models may eventually be used to model biological processes, such as the chemistry of nerve impulses.

Philippe Plailly / SCIENCE PHOTO LIBRARY

complex systems; to model the effect of perturbations or designed changes upon them; and to make or to change complex systems in desired directions. This three-element approach – Measure, Model, Manipulate – has delivered much of the progress in technology seen through the industrial revolutions, and remains spectacularly successful as an approach to the systematic delivery of technology today. Systematic approaches of this kind are to this point difficult to apply directly to highly heuristic systems such as biological systems. However, both measurement technologies and the predictive value of modelling are currently improving rapidly, putting in view the prospect of systematic approaches to biology, in which biological systems themselves may be engineered.

Should coherence increasingly emerge from complexity, it will leave no area of applied science untouched. Systematic approaches, enabled by an ability to predict, always trump heuristic ones, not least because they are less costly. We believe that the elements of a more systematic and less heuristic biology are already visible in part, and that a systematic ability to change, fix, or make biological and other complex systems themselves will be an important part of the scientific landscape by 2020.

Malcolm Young, Ehud Shapiro

[3] A helpful insight is that the Information in such a matrix is proportional only to the log of the different values that an individual entry may take. Hence, for small systems, much Information is typically present in the individual entries, and in the exquisite details of bilateral interactions between system elements, but, as *N* rises, for large systems with structured interactions, almost none is. For large *N* systems, almost all Information can lie in the topological and statistical distribution of interactions in the system.

New Conceptual and Technological Tools

The invention of key new conceptual tools (e.g. calculus) or technological tools (e.g. the telescope, electron microscope) typically form the building blocks of scientific revolutions that have, historically, changed the course of history and society.

Such conceptual *and* technological tools are now emerging at the intersection of computer science, mathematics, biology, chemistry and engineering. Indeed, as we shall see, we look set to be in extremely exciting and potentially profoundly important times towards 2020 as advances continue to be made in the development of these tools.

On the following pages we briefly outline some of the tools most clearly emerging now. There will be others that are created towards 2020 that we cannot imagine at the moment.

Codification of Biology

The next 14 years will see major activity in the codification of scientific knowledge. By *codification* we mean quite literally turning knowledge into a coded representation, in terms of data or programs, that is mechanically executable and analysable. The overall task often involves building mathematical models of natural phenomena, but goes beyond that, to the actual task of turning those models into coded representations that are useful to a whole community of scientists. Codification has already been undertaken in a number of areas, but it is just beginning in several major fields of scientific knowledge.

The field of *computing*, for example, can be described as the (development and) codification of information processing knowledge. That is, turning information processing architectures (e.g. algorithms) into executable and analysable information processing engines. Both hardware and software realisations are forms of codification.

The codification of mathematics has a long history, from logical and foundational studies to the current attempts to completely mechanise major mathematical proofs. Moreover, many fundamental numerical and symbolic techniques are already codified in mathematical libraries and tools.

Codification has at least one basic scientific property: once obtained, it can be right or wrong, or 'not even wrong', but it is at least exactly reproducible and independently analysable.

Biology is one area where codification is seen as crucial to general scientific progress. At the conceptually simplest level, we have the codification of the genome: DNA structures from various organisms are represented as long strings in a 4-letter alphabet. Once information is stored this way, it can be searched, compared, and analysed, using a wide variety of computational techniques. The next hardest task is the codification of the proteome. In that case, we need to store data structures that are significantly more complex: they are 20-letter strings (of amino acids) plus three-dimensional positional information, and various auxiliary annotations. This kind of representation is now relatively standardised, and several tools are available to take advantage of it.

Further efforts involve the codification of metabolic and signalling pathways. In that case, what needs to be stored, searched, compared, analysed, etc., are *networks* of biochemical interactions. How to do that is still pretty much an open problem, although many efforts are under way and many pathway databases are being created.

The general, hardest, problem in this area is going to be how to store, search, compare and analyse *biological processes*. A process, here, is intended as a dynamic interaction of multiple discrete components, e.g. the process of cell division. Even finding a standardised way of storing such information is a highly non-trivial task.

This last example brings into focus the full meaning of codification: it is not, in general, just to represent scientific facts as *data*, but to represent scientific

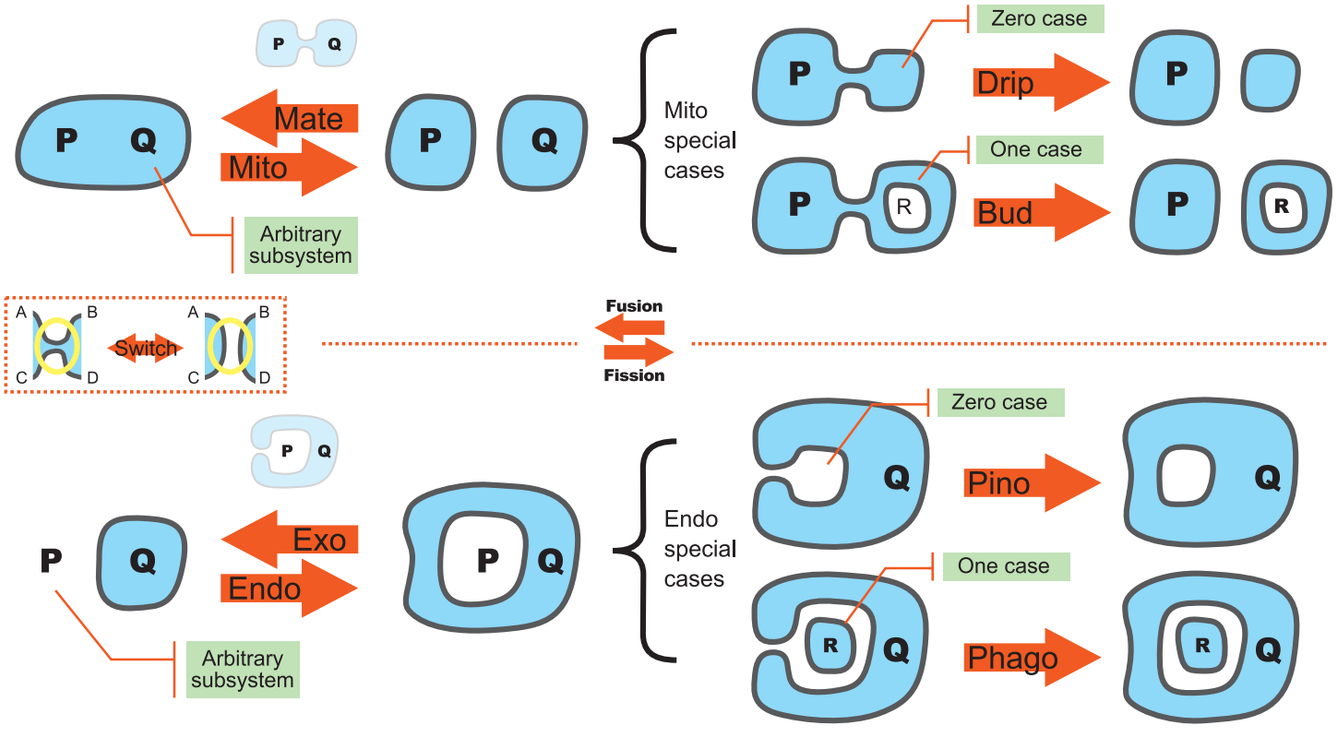
phenomena as dynamic *processes*, which need to be stored, searched, compared and analysed as such. Martin H. Fischer's aphorism "Facts are not science, as the dictionary is not literature" emphasises a similar point: codification of knowledge does not just mean putting all known facts into a large database; the facts must be organised into coherent 'literature' that itself must be storable and analysable.

Codification is largely an engineering enterprise: there are numerous issues about the best way of representing any given piece of information. While mathematicians and physicists tend to think in terms of structures that 'exist' in some real or imagined world, computer scientists (and logicians) tend to think of different *representations* of structures that (may or may not) exist. For example, there can be many ways, and endless discussions, about the best way to implement a given mathematical function. Similarly, for data, major efforts are undertaken to agree on standardised ways to represent basic facts.

Computing has developed a number of generally applicable engineering principles for effective codification: abstraction, compositionality, reusability, scalability, etc. These should be intended as engineering principles in a broad sense, where the measure of effectiveness can be computational efficiency, or can be effectiveness in organising and carrying out mathematical analysis. Computing techniques that work on coded information have been designed to be scalable, and are routinely applied to the most complex engineered (non-natural) systems in existence: software systems.

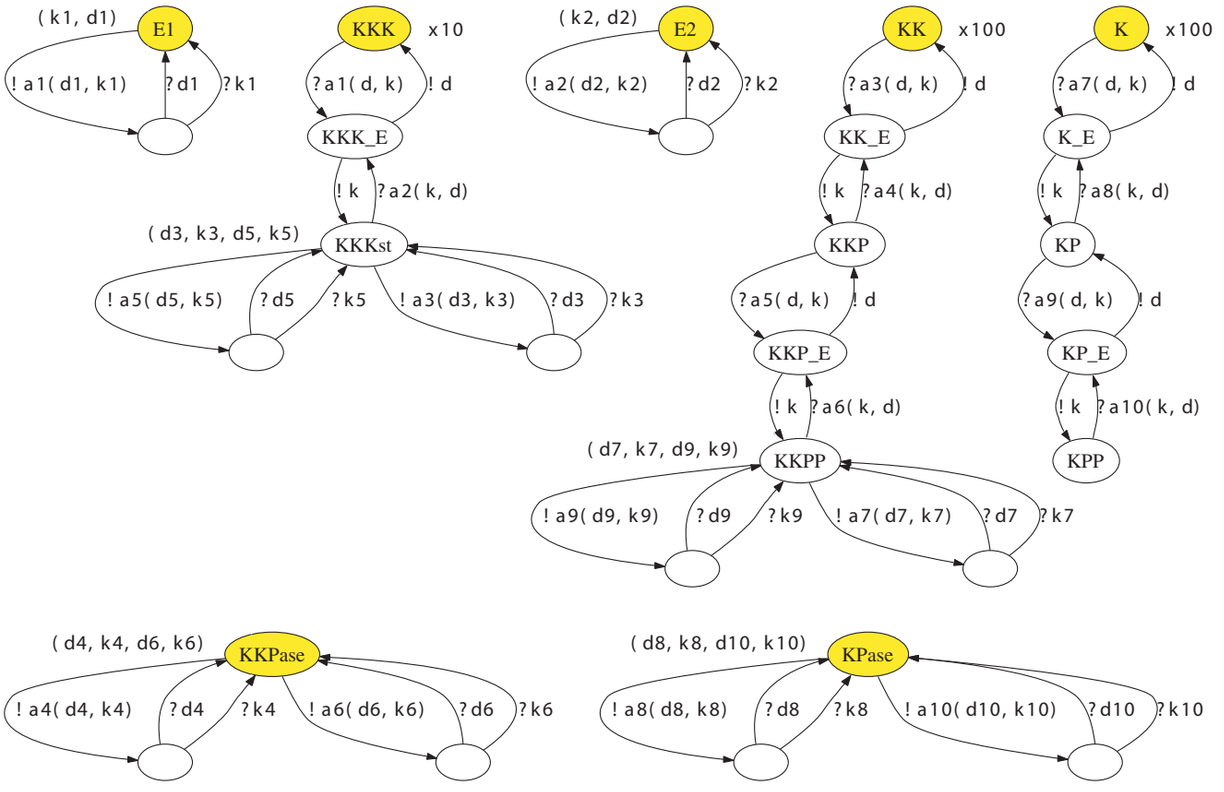
There are still great difficulties ahead in codifying scientific knowledge, particularly in the experimental and descriptive sciences, and in areas that deal with complex systems and with information-processing systems (with biology as a prime example [20]). Those are the areas where classical mathematical analysis techniques find difficulties in applicability or in scalability. It is our belief that principles and techniques from computer science will become fundamental in those areas, complementing current mathematical techniques, and that they will be successfully applied, in general, to the codification of scientific knowledge.

Luca Cardelli, Ehud Shapiro, Andrew Phillips



The Membrane Machine Instruction Set

The basic operations on membranes, implemented by a variety of molecular mechanisms, are local fusion (two patches merging) and local fission (one patch splitting in two). In two dimensions at the local scale of membrane patches, fusion and fission become a single operation, a switch. A switch is a fusion when it decreases the number of whole membranes, and is a fission when it increases such number. When seen on the global scale of whole 2D membranes, a switch induces four operations: in addition to the obvious splitting (Mito) and merging (Mate) of membranes, there are also operations, quite common in reality, that cause a membrane to 'eat' (Endo) or 'spit' (Exo) another subsystem (P). There are common special cases of Mito and Endo, when the subsystem P consists of zero (Drip, Pino) or one (Bud, Phago) membranes. Although this is an unusual computational model, the membrane machine supports the execution of real algorithms. In fact, some sets of operations, such as {Pino, Phago, Exo} are Turing-complete, and can encode the other membrane operations.



The MAPK signalling cascade

A stochastic pi-calculus model of the mitogen-activated protein kinase (MAPK) cascade [22] which illustrates how tools for codifying computer systems can also be used to codify biological systems. Each connected graph in the figure describes the behaviour of a protein in the cascade, where each node represents a protein state and each arc represents a potential interaction with another protein [23]. The stochastic pi-calculus can be used to model very large systems incrementally, by composing simpler models of subsystems in an intuitive way. The calculus also facilitates mathematical reasoning, which could help provide insight into some of the fundamental properties of biological systems.

Prediction Machines

The value of a scientific theory is determined in large part by its ability to make predictions. Quantum electrodynamics, for example, allows the anomalous magnetic moment of the electron to be calculated to more than 10 significant figures, in agreement with comparable experimental observations. Similarly, in applied science and technology, the ability to make predictions, for example, the biological activity of a candidate drug molecule, or the structural and aerodynamic properties of the wing for a new airliner, is again of central importance.

In some domains, the underlying equations are well understood, and in principle, predictions simply involve solving these using appropriate numerical techniques. Even here, however, many research questions often arise due to issues such as efficiency, numerical accuracy and model validation. Performance improvements in computers continually extend the range of predictions which can be obtained by numerical solution from first principles. In chemical dynamics, for instance, computational methods are having a major impact on the capabilities for first-principles prediction. Currently, reaction cross-sections can only be calculated for systems involving a few atoms at a time, and for each additional atom the computational cost increases by two or three orders of magnitude. While this clearly does not scale directly to large systems, the possibility exists to exploit the locality of chemical reactions so that only the handful of atoms directly involved in a reaction need be modelled in detail, with the remainder treated semi-classically, or quasi-classically, thereby opening the door to accurate modelling of complex chemical dynamics on computers having perhaps just a few hundred teraflops of processing power. If such an approach can be developed, the potential impact not only on chemistry but on neighbouring fields could be profound. There is no shortage of other drivers for large-scale simulation of basic physical processes. For instance, modelling of turbulent flows has motivated the development of sophisticated multi-scale techniques, while future grand challenges such as space weather forecasting, or tsunami prediction based on real-time computer simulation driven by inputs from networks of distributed sensors, could potentially lead to new breakthroughs.

It is increasingly easy to write simulation models, and these are intuitively more attractive to the non-mathematically inclined because of their less reductionist character, but arguably they require broad as well as deep mathematical expertise (all the way from statistics and numerical analysis to stochastic process theory and non-linear dynamics) to be applied correctly. Heuristically constructed simulation models, while sometimes producing intriguing results, are next to useless scientifically as the source code is rarely published (and is hard to interpret if it is) leaving at best a qualitative verbal description of the model's structure. Consequently, there is a pressing need to maintain mathematical and scientific rigour, as well as to ensure that models and their implementations are appropriately validated.

However, for most areas of science, the complexity of the domain, or the absence of sufficiently precise models at the appropriate level of description, often prohibit a first-principles simulation with any current or conceivable future level of computational resource. In such cases statistical approaches, in

particular machine learning, have proven to be very powerful. While classical statistics is focused on the analysis of data to test hypotheses, the goal of machine learning is to use (primarily) statistical methods to make predictions. For instance, in the basic supervised learning scenario, a large number of input-response pairs are used to construct a model of the input–output relationship of a system, capturing the underlying trends and extracting them from noise. The data is then discarded and the resulting model used to predict the responses for new inputs. Here, the key issue is that of generalisation, that is the accurate prediction of responses for new inputs, rather than simply the modelling of the training data itself. Machine learning techniques are also used for data visualisation, data mining, screening, and a host of other applications. For instance, in the area of biological modelling, techniques from Inductive Logic Programming (a form of machine learning which represents hypotheses using logic) have been demonstrated on a variety of tasks including the discovery of structural principles concerning the major families of protein folds [24], prediction of structure-activity relations in drugs [25] and prediction of toxicity of small molecules [26]. In addition, Bayesian networks, whose structure is inferred from observed data, have been used to model the effects of toxins on networks of metabolic reactions within cells.

Research into algorithms and methods for machine learning provides insights which can inform research into one of the greatest scientific challenges of our time, namely the understanding of information processing in biological systems including the human brain. For instance, in low-level visual processing, there are interesting similarities between the localised responses of cells in the early visual cortex and the wavelet feature bases which are found to be very effective in computer vision problems such as object recognition. More speculatively, the current interest in hybrids of generative and discriminative models in the machine learning field offers potential insights into the brain's remarkable ability to achieve accurate generalisation from training data which is almost entirely unlabelled.

Ongoing developments in machine learning over the last 5 years have significantly increased the scope and power of machine learning. Three such developments in particular, have been pivotal, namely the widespread adoption of a Bayesian perspective, the use of graphical models to describe complex probabilistic models, and the development of fast and accurate deterministic techniques for approximate solution of inference and learning problems. The Bayesian networks mentioned earlier are a particular instance of graphical models.

Machine learning techniques are not, however, confined to the standard batch paradigm which separates the learning phase from the prediction phase. With active learning techniques, the adaptation to the data and the prediction process are intimately linked, with the model continually pointing to new regions of the space of variables in which to collect or label data so as to be maximally informative. Indeed, as reported recently in *Nature*, an active learning framework was used for choosing and conducting scientific experiments in the Robot Scientist project (see following section on 'Artificial Scientists').

The two approaches to prediction, based, respectively on first-principles simulation and statistical modelling of observed data, need not be exclusive, and there is undoubtedly much to be gained in addressing complex problems by making complementary use of both approaches. In population biology, for example, a complete treatment is likely to require combination of elements from non-linear dynamics, complexity science, network theory, stochastic process theory and machine learning.

Many of the developments in the computational sciences in recent years have been driven by the exponential increase in the performance of computer hardware. However, the limits of the single processor are already being reached, and in order to sustain continued exponential growth, the manufacturers of processors are already moving towards massively multi-core devices, posing some major challenges for software developers. Fortunately, many machine learning algorithms, as well as a significant proportion of numerical simulation methods, can be implemented efficiently on highly parallel architectures. Disk capacity, as well as the size of scientific data sets, have also been growing exponentially (with a shorter doubling time than for processors and memory) and so the need for effective data mining and statistical analysis methods is becoming greater than ever. Coupled with the continuing developments in models and algorithms for machine learning, we can anticipate an ever more central role for statistical inference methods within much of the computational science arena. As probabilistic inference methods become more widely adopted, we can anticipate the development of new programming languages and user tools which embrace concepts such as uncertainty at their core, and which thereby make the process of implementing and applying machine techniques substantially more efficient, as well as making them accessible to a much broader audience.

Christopher Bishop, Stephen Muggleton, Aron Kuppermann, Parviz Moin, Neil Ferguson

Artificial Scientists

As a consequence of the scale and rate of data generation in science, of which some examples were outlined in Part 1, models of the data are increasingly requiring automatic construction and modification. Moreover, it is already clear that computing and computer science are set to play an increasingly central role in supporting the fundamental formulation and testing of scientific hypotheses. This traditionally human activity has already become unsustainable in the biological sciences without the aid of computers. This is not primarily due to the scale of the data involved but is because scientists are not able to conceptualise the breadth and depth of the relationships and potential relationships contained within the data.

Machine Learning systems that produce human-comprehensible hypotheses from data will increasingly be used for knowledge discovery within science. Such systems today are typically open loop, with no direct link between the machine learning system and the collection of data. A more closed-loop approach was investigated in the early 1990s in work on automating chemical experiments [27], though the approach was limited to the estimation of chemical parameters. However, recent advances in computer science go considerably beyond such approaches, pointing the way to an exciting and intelligent future – one of 'autonomous experimentation'. In the new approach, artificial intelligence techniques are employed to carry out the entire cycle of scientific experimentation, including the origination of hypotheses to explain observations, the devising of experiments to test these hypotheses and the physical implementation of the experiments using laboratory robots to falsify hypotheses. Such a system has already been demonstrated in the 'Robot Scientist' project [28] where laboratory robots conducted experiments selected by 'active learning'.

The Robot Scientist project demonstrated that using a machine learning system, based on Inductive Logic Programming (ILP), the robot selected experiments to discriminate between contending hypotheses. The experiments were based on gene knock-outs for yeast (*Saccharomyces cerevisiae*). The aim was to determine the function of the gene by varying quantities of nutrient provided. Feedback on the outcomes of experiments came in the form of indications of whether or not the yeast died within a 24-h period. The robot's intelligent experiment selection strategy based on the ASE-Progol system was competitive with human performance and significantly outperformed both cheapest and random-experiment selection with a cost decrease of 3-fold and 100-fold, respectively.

One exciting development we might expect to see in this area over the next 10-years is the construction of the first micro-fluidic Robot Scientist. This would involve the confluence of active learning and autonomous experimentation systems with micro-fluidic technology (see section on 'Molecular Machines' below). The key effects of miniaturising the technology would be the reduction of the experimental cycle time from hours, in the case of the previously published Robot Scientist, to milliseconds, with a corresponding increase in the robustness of outcomes from micro-fluidic experiments. Further flexibility could be added to such a scenario by the use of Chemical Turing machines (see next section) to allow automatic online preparation of a wide variety of chemical compounds as input to each experiment.

Conducting impossible experiments

Autonomous experimentation will undoubtedly play an important role towards 2020 in meeting the challenge of accumulating and analysing the comprehensive data sets (e.g. such as is required for biological modelling at the system level) that are outside 'normal' affordable resource constraints including time and human bandwidth.

Moreover, such computational techniques capable of deciding from past observations which test to perform next are also of great interest for instrumentation with limited communication bandwidth where decisions need to be taken remotely as to which is the best 'next step' to take, such as which experiment to perform, when to perform it or when to stop collecting data. Mobile robots exploring remote or harsh environments such as deep sea locations or other planets typically are able to communicate back only a small fraction of the data gathered by their sensors. Accordingly, the robot itself would be in the best position to decide the next experimental step.

In the coming decade, we can expect a confluence of wireless networks and lab-on-chip sensor technology with large-scale applications in environmental monitoring. Autonomous experimentation will be required to fully exploit the potential of this technology. In such a lab-on-chip network, each sensor node is endowed with a limited supply of wet chemistry and accordingly can perform only a limited number of experiments. The network will collectively decide how these resources should be spent.

In summary, autonomous experimentation enabled by intelligent 'Artificial scientists' has begun to open up highly novel experimental approaches that have previously been unimaginable.

Stephen Muggleton, Klaus-Peter Zauner

Molecular Machines

The distinction between the 'artificial' and the 'living' is being increasingly blurred by new technological advances at the intersection of computing, biology, chemistry and engineering. These advances have the potential to revolutionise not only our ability to model, understand and repair complex living systems, but also to construct new biological parts that may one day be the building blocks of entirely new organisms. This is a world made possible by labs-on-chips, synthetic biology and molecular computers. The implications are truly profound, not only for biology and medicine, but also for human ethics and for protecting our society against future onslaughts of natural (and engineered) diseases.

Labs on Chips

Living cells are the most sophisticated nano-systems known. The speed with which they reproduce belies their intricate intracellular organisation. This organisation gives rise to a very special environment, whose physics is characterised by small length scales, surfaces dominating volume, laminar flows, fast diffusion, and the heat bath, and whose chemistry is dominated by stochastic fluctuations and highly controlled reactions. Recent progress in microfluidics and nanotechnology has now opened this unfamiliar environment to practical engineering experience [29]. Microfluidics has been used to construct systems of channels and vessels equipped with electrodes and optics for detecting, measuring, and manipulating solutions, particles, macromolecules or cells. Simple laboratory work flows, such as sequences of reactions followed by product analysis, can be implemented on a single chip for mass production. Such micro-reaction chemistry within confined volumes offers unprecedented control over reaction conditions. In addition, specific macromolecules or cells can be individually identified and separated using on-chip methods, enabling new types of experiments to be conducted. Since the reactions are in small volumes, they consume very little chemical supplies, enabling extensive studies. Furthermore, high integration on a single chip enables new types of instrumentation that use embedded chemistry to perform analysis at the point of measurement.

Synthetic Biology

Although lab-on-chip technology allows for refined control of molecular interactions, it pales in comparison to the powerful infrastructure and mass production capabilities of a living cell. For over two decades, efforts have been under way to modify cells for factory production of chemicals that are either too complex for classical synthesis, or that can be produced much more efficiently by microorganisms (white biotechnology). So far, most of these examples have been relatively simple and could probably have been identified by classical mutagenesis. However, as genome-wide computational models become more complex, these models will enable the construction of cell factories that could change the very nature of production in the chemical and pharmaceutical industries. The self-replication capability inherent to cells lends itself to convenient mass production. More recently, there has also been a shift from static metabolic engineering towards interfacing with the control structures of cells [30]. The possibility to engineer the dynamics of cellular behaviour opens a path to novel, living biomaterials.

Cells tuned for a particular purpose can be grown in bio-films of communicating elements, going far beyond the conventional idea of DNA computers [31]. This is leading to the emergence of the field of *Synthetic Biology*, focused on 'the design and construction of new biological parts, devices, and systems, and the re-design of existing, natural biological systems for useful purposes'. MIT is leading the evolution of this new field through the BioBricks project (http://parts.mit.edu/), with the development of standard and interchangeable biological parts. These currently include operators, coding regions, transcriptional terminators and logic gates. The aim is to provide powerful abstractions so that bio-engineers can design and build complex biological systems in the same way that electronic engineers have traditionally built complex electronic circuits, by putting together high-level logic gates.

Bio-hybrid chips

Customised cells can also be integrated with conventional technology, and integrated circuits have already been interfaced with microorganisms engineered for sensing [32]. This technology enables the sensing and discrimination of minute traces of a substance (e.g. a toxin) against a complex chemical background. Lab-on-chip technology is capable of maintaining the environment required by the cells resident in such a bio-hybrid chip. The integration of microorganisms into electronic circuits can be expected to expand significantly within the next 5 years.

Molecular computing

Cell simulation through building such artificial cells is viewed as a widely desirable goal by systems biologists. However, an alternative engineering approach would be to design a self-replicating von Neumann machine – natural and artificial cells can be considered as a special case of a von Neumann Universal Constructor [33]. A chemical synthesis version of the constructor has already been described by Drexler [34], and ideas for miniaturisation of machines were discussed still earlier by Penrose and Penrose [35] and Feynman [36]. These can all be viewed as special cases of a *Chemical Universal Turing Machine* (CUTM) – an abstract computational device used to investigate what can be computed. It should be possible to build such an abstract machine with a simple physical realisation. This would be a simple automaton consisting of a large reaction flask, together with a conveyor belt containing an arrangement of chemicals. The chemicals would constitute the 'program' and the automaton would have operations for reading chemicals on the conveyor, adding and removing them from the reaction flask, and controlling the temperature of the reaction. The CUTM is a fundamental concept which unifies lab-on-chip and artificial cell concepts. Clearly, this requires research to establish its viability but, if successful, could have a dramatic effect on combining theoretical, experimental and modelling approaches to understanding and engineering biology.

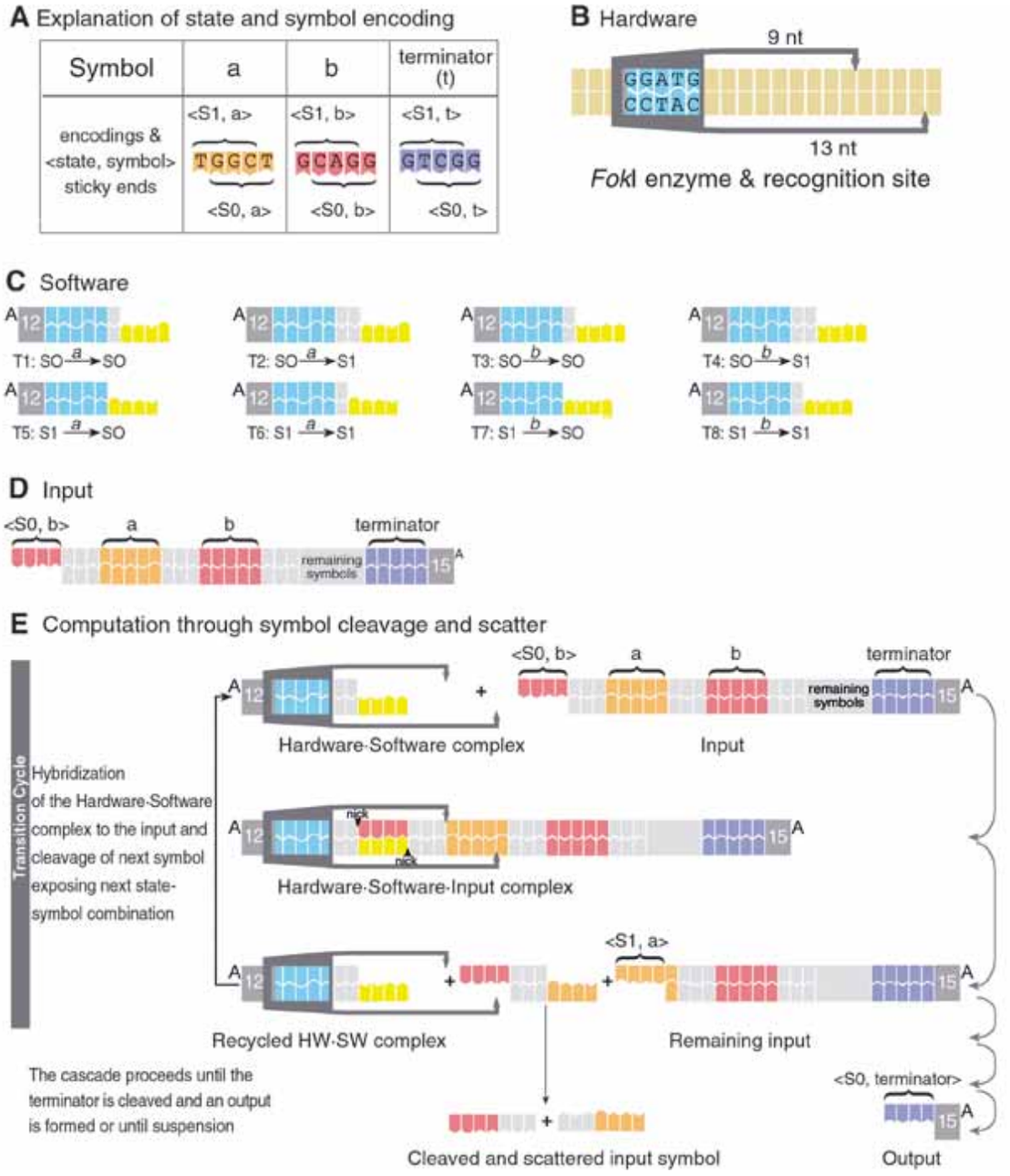
Much closer to realisation than self-replicating machines are simple molecular computers. Comparable in capability more to a laundry machine controller than to a general purpose computer, simple molecular computers are small enough to operate within a cell. A proof-of-concept of a molecular computer has recently been demonstrated, which senses disease-related molecular symptoms, namely over – and under-expressed genes, analyses the information according to a medical rule and, if a disease is diagnosed, releases the drug molecule it carries [37]; see also the section 'Revolutionising Medicine' in Part 3. This concept is interesting, not least because it illustrates the potential of combining key concepts from computer science with chemistry and biology into a new form of therapy: smart drugs for nano-medicine. In this case, the computer has three modules: an input module that senses molecular symptoms, a computation module that analyses symptoms according to pre-programmed medical knowledge, and an output module that releases the drug. To offset for unreliability of computer components, two types of automata compute in parallel: one that releases a drug upon positive diagnosis, and one that releases a drug suppressor upon negative diagnosis. Many hurdles are to be overcome in developing this into a viable molecular computer for applications in areas such as smart drugs, yet even if this design does not cross these hurdles, it might still inspire future designs that would. The consequences would be dramatic – revolutionising biology and medicine.

Summary

Insights from systems biology combined with an understanding of molecular mechanisms will increasingly enable the tailoring of cells for specific purposes. On a long-term perspective, the possibilities created by the emerging tool set of synthetic biology are enormous. They span from new experimental tools and techniques to the design and synthesis of entirely new forms of drugs [38]. A detailed understanding of the information processing principles operating within cells will be crucial to realising these possibilities.

With the simplification of cells to their minimal set of essential genes [39], the bottom-up implementation of micro-compartmentalised biochemistry [29] and the integration of cells into circuits [32] well under way, we can expect the distinction between biological and technical systems to blur in the coming decade. The computer will be central in this merger of 'the artificial' and 'the living'. Principles from computer science are already proving essential for addressing the immense complexity of such an endeavour.

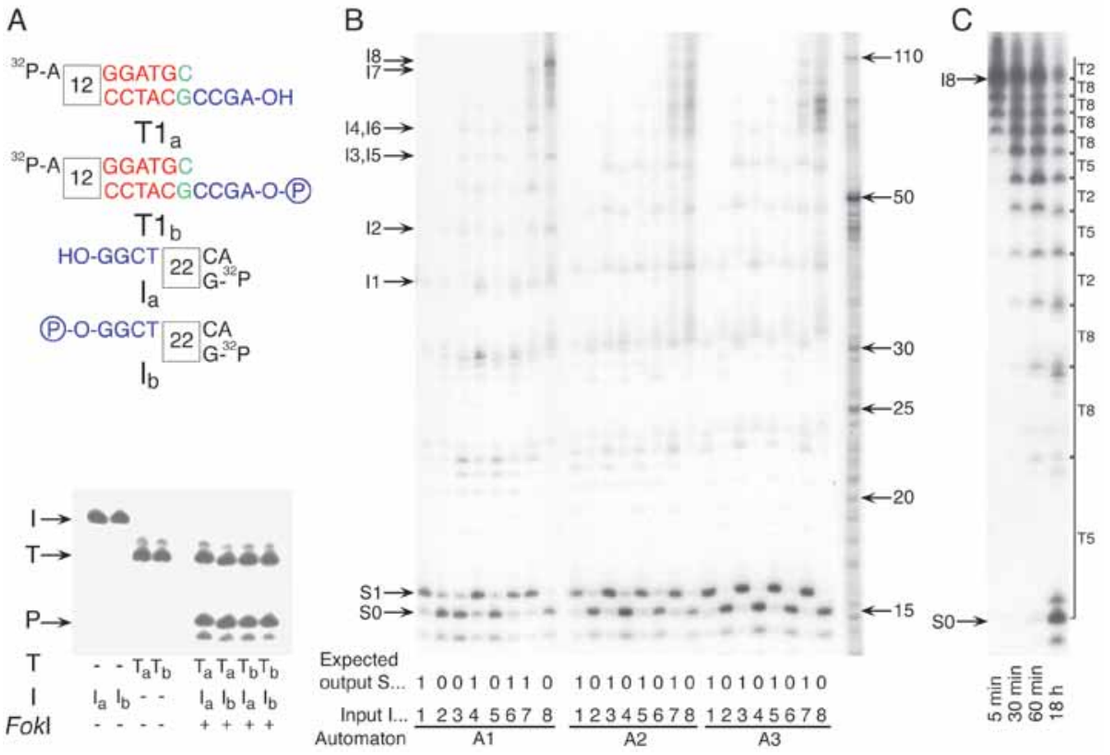
Klaus-Peter Zauner, Søren Brunak, Andrew Phillips, Ehud Shapiro, Stephen Muggleton



A molecular finite automaton that uses input as fuel

(A) Encoding of a, b, and terminator (sense strands) and the <state, symbol> interpretation of exposed 4-nt sticky ends, the leftmost representing the current symbol and the state S1, similarly the rightmost for S0. (B) Hardware: The *Fok*I restriction enzyme, which recognises the sequence GGATG and cleaves 9 and 13- nt apart on the 5'→3' and 3'→5' strands, respectively. (C) Software: Each DNA molecule realises a different transition rule by detecting a current state and symbol and determining a next state. It consists of a <state, symbol> detector (yellow), a *Fok*I recognition site (blue), and a spacer (grey) of variable length that determines the *Fok*I cleavage site inside the next symbol, which in turn defines the next state. Empty spacers effect S1 to S0 transition, 1-bp spacers maintain the current state, and 2-bp spacers transfer S0 to S1. (D) Input: The exposed sticky end at the 5' terminus of the DNA molecule encodes the initial state and first symbol. Each symbol is encoded with 5-bp separated by 3-bp spacers. (E) Suggested mechanism of operation of the automaton. The computation proceeds via a cascade of transition cycles, each cleaving and scattering one input symbol, exemplified with the input molecule bab in the initial state S0 and the transition S0 $\xrightarrow{b}$ S1. Both hardware and software molecules are recycled.

Benenson Y. *et al*. Proc. Natl. Acad. Sci. U.S.A. 100(5): 2191 – 2196.
Reproduced with permission.



Experimental results and mechanism analysis of the molecular automaton

(A) A demonstration that a computation does not require ligase. Different variants of the software molecule T1 ($T1_a$, nonphosphorylated, and $T1_b$, phosphorylated) and the input (Ia, nonphosphorylated and Ib, phosphorylated) were incubated with the hardware (*Fok*I) at 8°C for 10 min. Input, software, and hardware concentrations were all 1 μM. Reaction components are shown below the lanes, and the locations of software molecule (T), input (I), and product (P) are indicated by arrows. (B) Executing automata A1-A3 on inputs I1-I8. Input, software, and hardware concentrations were 1, 4, and 4 μM, respectively. Reactions were set in 10 μl and incubated at 8°C for 20 min. Each lane contains inputs, intermediate configurations, and output bands at the indicated locations. The programs, inputs, and expected outputs are indicated below each lane. Location of unprocessed input is shown on the left. Size markers are indicated by arrows on the right. (C) Software reusability with the four-transition automaton A1 applied to input I8 with each software molecule taken at 0.075 molar ratio to the input. Input, software, and hardware concentrations were 1, 0.3 (0.075 μM each kind), and 1 μM, respectively. After 18-h, molecules T2, T5, and T8 performed on average 29, 21, and 54 transitions each. Input and output locations are indicated on the left, and intermediates and software molecules applied at each step are on the right.

Benenson Y. *et al* . Proc. Natl. Acad. Sci. U.S.A. 100(5): 2191 – 2196.

New Software Models for New Kinds of Tools

As numerous sections of this report make clear, science will become increasingly reliant upon new kinds of tools towards 2020 – and increasingly highly novel software-based tools. As a result, new approaches to the development of software-based tools for science will be required to enable important scientific breakthroughs and enable scientists to be productive. At the same time, scientists will be required to be increasingly computationally competent in ways that benefit the overall scientific community. Both of these demands present considerable software challenges. How these challenges can be addressed requires a consideration of trends and developments in a number of areas discussed next.

Software engineering

Single processors with uniform memory systems and hierarchies are being replaced by non-uniform multi-processor ('multi-core') systems that defy present programming models. Scaling already challenged by huge datasets is now also challenged by dramatically more complex computing platforms, previously only known to supercomputing. The challenge is multiplied by the need to integrate across the Internet. The implications for science are potentially enormous. Concurrent programming on multi-core machines is likely to mean scientists become more reliant on software platforms and third party libraries to accrue the benefits of this processing power. Validating computational results will also become more difficult, as non-determinism may become normal for complex calculations. Combining the trend towards non-uniform, parallel hardware with the computational needs of science (perhaps more than any other area) to draw on the limits of hardware, and the need to embrace ever-more complex approaches leads to a tough challenge: to devise models enabling robust design and development of software components and frameworks and the flexible composition of components, in combination with ad-hoc code, to address the needs of rapidly diversifying, converging and evolving sciences. This is a challenge for software and software engineering.

We need new programming models to cope with such new hardware topologies. This will require coordinated scientific efforts to build sharable component frameworks and matching add-in components. New software architecture methods, especially to support factoring and recomposition (to enable effective sharing among collaborating teams or even within and across disciplines) require an emphasis on components, component frameworks, component technologies, properly supported by architectural concepts.

Componentisation

Software presents itself at two levels: source and executable. Sharing source has its advantages, but as a unit of sharable software, source-code fragments are too brittle. A solid concept of software components is instead required where components are viewed as units of deployment. A deployable unit is something sitting between source code and installable, executable code enabling the delivery of software components parameterised for a range of scenarios. Components can also encapsulate expertise – not everyone using a component would have to be

capable of developing an equivalent component. In the context of science, it is most compelling to consider both arguments: efficient sharing of components and encapsulation of expertise to leverage complementary skills. Shared use of components in an unbounded number of compositions enables a systematic approach to the support for evolving scientific projects.

It is critically important to understand the boundary conditions that enable composability. Components only compose in the context of a component framework [40,41] – reference standards that establish strong conditions for integration such that components designed to meet a component framework, even if developed by mutually unaware parties, will compose.

Software services

Providing specialised software services (such as up-to-date genome catalogues) is compelling in science. It is conceivable that both government and industrial funds will help maintain such services, but the absence of a simple 'business model' leads to a reliance on some form of sponsorship. The actual sharing of computational resources (in a way, the most basic software service that can be offered), as envisaged by some Grid projects, seems less compelling. There are a few examples (like the SETI@Home project) that lend themselves to this approach since the underlying computation is parallel and the data sets that need to be distributed are relatively small. In most cases, however, computational resources are cheap and widely available. It would therefore seem that the sharing of services is much more interesting than that of computational resources.

Software engineering

Software engineering for science has to address three fundamental dimensions: (i) dealing with datasets that are large in size, number, and variations; (ii) construction of new algorithms and structures to perform novel analyses and syntheses; and (iii) sharing of assets across wide and diverse communities.

Algorithms and methods need to be developed that self-adapt and self-tune to cover the wide range of usage scenarios in science. In addition, there is a need to develop libraries of componentised assets that can be generated for a broad spectrum of platforms, preferably targeting software platforms (such as managed-code platforms) that shield software assets from a good part of the underlying hardware variety. In many cases, supporting *little languages* that match the domains of particular component libraries can help reduce what would be complex problems to natural programming tasks. Platforms that integrate across a broad range of languages and tools can enable such an approach.

To move beyond applications and enable broader integration, service-oriented architectures are required: applications need to be built such that they can mutually offer and draw on each other's services. However, applications that are temporarily disconnected (partially or totally) need to continue offering some autonomous value. This is critically important to avoid unmanageable dependences across loosely coordinated research organisations. Service orientation can also enable the moving of computational activity to where datasets reside; a strategy that, for huge datasets, is often preferable over the traditional approach to move datasets to where computation takes place.

Programming platforms

Managed platforms

A significant trend in software development for the last 10 years has been the move from programming languages to programming platforms, exemplified primarily by Java™ and the Microsoft® .NET™ Framework. These 'managed platforms' encompass:

- platform-oriented languages (e.g. Java™, Visual C#® and others);
- a virtualised, high-performance, secure runtime engine;
- base libraries suitable for modern programming tasks such as concurrency, distribution and networking;
- key abstractions enabling extensibility and componentisation;
- visualisation, graphics and media engines;
- integration with related technologies such as databases and servers;
- a range of related software design and analysis tools;
- support for a range of interoperability formats.

Managed platforms dominate commercial programming and is a trend we expect to continue to grow in science, and indeed the dominance of managed code in science is both desirable and almost certainly unavoidable. Notwithstanding this, many niche areas of software development exist where alternatives and/or enhancements of managed platforms are deployed and used by scientists, including Python, Perl, Scheme, Fortran, C++, MATLAB® (by MathWorks; http://www.mathworks.com) and the language R (http://www.r-project.org/).

A key feature of managed platforms is that they combine multiple approaches to compilation and execution, enabling the platforms to be configured for a range of development and deployment tasks. Increasingly, managed platforms are also heterogeneous in the sense that many different kinds of programming are addressed by the platform; examples include:

- Domain-specific embedded languages for computation (e.g. utilising graphics hardware for matrix computations)
- Mathematics-oriented, scalable, script-like programming languages (e.g. F#; http://research.microsoft.com/projects/fsharp)
- Interoperable scripting languages (e.g. Iron Python, JPython)
- High-quality interoperability layers for existing environments and languages (e.g. MATLAB-to-Java connectivity, also Fortran and Ada for the .NET™ platform)

A key success of managed platforms has been to bring uniformity to kinds of programming that traditionally required *ad hoc* solutions. For example, it is remarkable that the same programming skills can now be applied throughout the components of a heterogeneous system, e.g. Visual Basic®, C# and Java™ may all be

used for client-side web programming, server-side programming, small devices and even within database processes. Hosting a computation at the right locale, such as inside a database, can yield major performance benefits (such as Microsoft's SQL Server 2005 Stored Procedures [42] and related functionalities in many other databases). This will inevitably result in significant portions of program execution being hosted on remote machines. To match this trend, development tools are also likely to be hosted increasingly.

Discoverability through visualisation during software construction

Many of the tools that make up software toolkits focus on ensuring programmers can quickly discover how to use software components. Visual Studio® (http://msdn.microsoft.com/vstudio/) and Eclipse (Eclipse: a kind of universal tool platform; http://www.eclipse.org) support a number of features to allow programmers to quickly discover and navigate through programmatic interfaces. The theme of discoverability now occupies much of the design effort in software development environments. Discoverability through interactive visualisation is likely to be critical for future scientific software development.

Correctness

Computer scientists have assumptions that place them in a difficult position when providing for the needs of scientists. One characteristic of scientific programming is that top-level code is 'write once, run once' (WORO). Even if a component-sharing community were established, such code will not evolve into published components. However, repeated use in multiple contexts is the foundation to 'correctness by reputation'. Computer scientists need to accept that highly complex code may be published primarily for the purpose of analysis and validation by other scientists, rather than for direct reuse. Software published primarily for validation will require non-traditional software-engineering techniques. Recent advances in formalised mathematics show how machine-checked mathematical results can be developed in a way where the result is independently checkable through simple means, even if the process of constructing a result was extremely challenging [43].

Scientists as programmers

Science is at the fore among the programming disciplines for the extreme demands it places on systems and software development. While tools such as spreadsheets are increasingly used for what are essentially 'end-user programming' tasks (a largely neglected trend among computer scientists [44]), on the whole, existing platforms and scientific programming environments are perceived to be suboptimal for science. While positive about some aspects of platforms, e.g. libraries provided by Java™, .NET™ and/or MATLAB®, common complaints centre around productivity of the scientist/programmer and the lack of a significant componentisation story. Licensing and legal considerations and fear of 'platform lock-in' are also a concern. Many scientists are clearly frustrated at the constraints placed on them by the software they have chosen to use, or by externally imposed constraints such as the need to use C++ in order to ensure their code can be added to existing applications. They are also very creative when it comes to working around limitations in interoperability.

The fundamentals of a good platform for the working programmer-scientist are clear enough: performance, ease of expression, scalability, scripting, an orientation toward mathematics, science-related libraries, visualisation, tools and community. Drawing on the success of MATLAB®, Mathematica®, spreadsheet systems, etc., it is useful to think of the solution to meet the requirements of science and scientists towards 2020 as a front-end that is something akin to an 'Office for Science': an extensible, integrated suite of user-facing applications that remain integrated with the development environment and that help address the many human-computer interaction issues of the sciences.

A new generation of advanced software-based tools will be absolutely critical in science towards 2020. Where scientists today rely on the creation of critical software assets as a side effect of general research, leading to large numbers of weakly maintained and mutually non-integrating libraries and applications, there is a strong need to form collaborative communities that share architecture, service definitions, services, component frameworks, and components to enable the systematic development and maintenance of software assets. It will be less and less likely that even the steepest investments focusing on specific projects will be leveraged in follow-on or peer projects, unless the software-engineering foundation of such efforts is rethought and carefully nurtured. Significant challenges for governments, educators, as well as scientific communities at large are paired with challenges of technology transfer and novel development of appropriate methods and technologies in the field of software engineering. Governing bodies will need to be established and properly funded that help with curation and perhaps coordination to have any hope of progress.

To empower communities of collaborating scientists across diverse organisations, appropriate methods and tools are required. Such tools will have to draw on rich metadata, encoding facts and knowledge, organised using appropriate semantic frameworks. Construction and support of loosely-coupled, collaborative workflows will enable specialists to collaborate on very large projects. Any such collaborative sharing will have to address issues of security, privacy, and provenance.

It is essential that scientists be proactive about ensuring their interests are addressed within software platforms. Platform providers must also recognise their unique responsibilities to the sciences, including a social responsibility to ensure the maximum effectiveness of the scientific community as it tackles the scientific problems of the coming century. A key challenge for scientists is to balance the tensions involved in guiding the design of tools on which they are dependent, including the need to (i) remain sufficiently distant from individual platforms in order to reap the benefits of innovation across a range of platforms; (ii) be deeply engaged in development plans for individual platforms to ensure that the requirements of a range of disciplines are catered for; (iii) be pro-active in standardisation efforts and in calling for interoperable solutions; (iv) communicate the peculiarly stringent requirements that science places on software platforms.

Science will benefit greatly from a strong, independent, informed, representative voice in the software industry, as will the broader communities served by science.

Finally, the trends outlined above will lead to major alterations in how we perceive software and the software construction process, allowing the opposite flow of innovation. As increasingly demanding and complex solutions in science are invented, the resulting solutions are likely to be transferable to the wider software space.

Clemens Szyperski, Wolfgang Emmerich, Don Syme, Simon Cox

New Kinds of Communities

New software-based tools will proliferate as they become increasingly essential for doing science. An inevitable consequence will be the combining by scientists of shared and differentiating software – an expression of the balance and tension between collaboration and competition that science benefits from. In the spirit of open science, differentiating software is ideally viewed as a contribution to the sciences and thus is eventually transferred to shared status. The high bar of demanding repeatability of scientific experiments equally demands the standardisation of instruments for all elements of the experimental method, parameter setting and estimation and data collection, collation and treatment – increasingly, this will not just include software, it will depend on software. This also means (i) the sharing of these 'tools' by and in the science community, (ii) their re-usability by others wishing to replicate or build upon the experiment(s) in which the tools were used, and (iii) their longevity, ensuring repeatability over prolonged time periods.

Thus, it would seem likely that the successful bootstrap of communities that build and share effectively at the level of components, frameworks/architecture, and services would follow a pendulum model, where contributions to the community first ripen in the more closed settings of a local group, followed by an effort to release the more successful pieces to the broader community, followed by more closed enhancements, further developments, and so on. Acknowledging the pendulum process embraces the competitive side of the sciences, with a clear desire of groups to be the first to publish particular results, while still benefiting by standing on the shoulders of the community collectively creating, developing and using scientific software tools.

A related challenge is educating computer scientists and software engineers to get such an approach off the ground [45]. Whatever works best for a professional software architect and engineer is not automatically what works best for the dedicated scientist who also 'does software', but it should provide guidance. Moreover, certain aspects of the approaches outlined in this report are hard and perhaps most effectively left to professional specialists. This is particularly true for the creation of sufficiently powerful and reasonably future-proof reference architecture and component frameworks. But this in turn requires a much deeper integration and/or relationship between computer science (and computer scientists, as well as software engineers) and the science community. Appropriate integration of professional support into the fabric of the science community is a challenge. Currently, support is typically that contributed by dedicated enthusiasts – often doctoral or post-doctoral students. However, where contributions from such efforts are meant to scale, be sharable, and to compose with efforts from others, there is a need to keep the framing straight and according to code.

In conclusion, it is clear that the computer science community and the science community need to work together far more closely to successfully build usable, robust, reliable and scalable tools for doing science. This is already happening in some areas and in some countries, but the scale of the integration required is not going to happen by accident. It will require a dedicated effort by scientists, numerous government initiatives to foster and enable such an integration, and the co-involvement of commercial companies such as Wolfram, MathWorks, IBM®, Apple® and Microsoft®. It is our recommendation that government science agencies take the initiative and introduce schemes and initiatives that enable far greater co-operation and community building between all these elements.

Clemens Szyperski, Stephen Emmott

3 Towards Solving Global Challenges

The 21st Century is already starting to present some of the most important questions, challenges and opportunities in human history. Some have solutions in scientific advances (e.g. health), while others require political or economic solutions (e.g. poverty). Some require significant scientific advances in order to provide the evidence necessary to make fundamental political and economic decisions (e.g. our environment).

Addressing any of them is non-trivial. In Part 3, we outline some of the primary global challenges for the first half of this century that we can at least foresee now that science can help to address, and how the advances in computing and computer science discussed in Parts 1 and 2 can accelerate advances in science in order to enable science to do so.

Deforestation

Rainforest being burned to clear land for cattle. Rainforest destruction may lead to countless plant and animal species becoming extinct. The loss of oxygen-producing vegetation also contributes to global warming due to increased levels of carbon dioxide in the atmosphere. As well as for ranching, rainforests are cut down for timber, to clear land for crops and for oil drilling. Photographed in Brazil.

Jacques Jangoux / SCIENCE PHOTO LIBRARY

Earth's Life-Support Systems

Authoritative assessments of the state of the Earth's life support systems -broadly speaking the 'biosphere' (biodiversity, ecosystems and atmosphere) – show major changes in their composition, structure or functioning [46,47]. For example, currently, human activity is producing 300% more carbon dioxide per year than the earth's natural carbon sinks can absorb [47] and this is expected to increase significantly over the next 2-3 decades at least as growth continues in developing countries such as China, India and South America. The result of this and other human activity is a potentially profound change in climate patterns and the consequent effects this could have. Moreover, we are losing a vital resource for life – the Earth's biodiversity – at a rate probably 100 times greater than from natural loss [46], and many of the Earth's natural resources are being grossly over-exploited. For example, 90% of Brazil's 100 million square kilometres of coastal forest, once one of the most diverse ecosystems on Earth, has been destroyed in the past 90 years, and fishing has massively depleted most of the world's fish populations in just a few decades. Perhaps most worrying is the Millennium Ecosystem Assessment [47] recent evidence that, out of the 24 'Life-support services' that nature provides and that we rely on for our continued existence, 15 are being used far faster than nature can regenerate them – and we should expect this number to rise still farther.

There is a fundamentally urgent need to understand the Earth's life support systems to the extent that we are able to model and predict the effects of continued trends of human activity on them, and the consequent effect on the ability of life to be sustained on the planet – including humans. This requires the development of extremely powerful predictive models of the complex and interacting factors that determine and influence our ecosystem and environment, and use these models to generate and evaluate strategies to counteract the damage the Earth is being subjected to.

Several areas of science are beginning to tackle this through integrating theory, remote sensing experiments and traditional observational studies, and computational models. Certainly, climatology and several branches of organismic biology (see below) depend increasingly upon computer science since their theories, which formerly were the sole province of mathematical biology, are becoming more and more computational in form [48].

Computational simulation and modelling of the dynamic behaviour of the world's climate are possible today, thanks to the efforts of numerous research centres, including the Earth Simulator in Japan and the Hadley Centre for Climate Prediction and Research in the UK. Climate modelling and prediction is an obvious and critical aspect of understanding the Earth, but we should anticipate being able to understand and model other key 'abiotic' systems. Focusing for example on the 'interior activities' of the planet, will influence our capacity to understand plate tectonics and geomagnetism, and by extension, our capacity to anticipate natural disasters such as earthquakes, volcanoes and tsunamis, perhaps as early as by 2010.

The other critical aspect of our understanding relies on knowledge of the 'biotic' Earth. Organismic biology, the study of biological entities above the cell level,

spans the ecology of single populations to that of the whole biosphere, and from micro-evolutionary phenomena to palaeontology, phylogenetics and macroevolution. Across this discipline, the number and size of databases (species observations, phylogenetic trees, morphology, taxonomies, etc.) are growing exponentially, demanding corresponding development of increasingly sophisticated computational, mathematical and statistical routines for data analysis, modelling and integration. The nascent field of *biodiversity informatics* – the application of computational methods to the management, analysis and interpretation of primary biodiversity data – is beginning to provide tools for those purposes [49].

Clearly, the biotic and abiotic need to be modelled together: at geographical scales, climate is one of the main determinants of species' distribution and evolution. Climate change will affect not only species' distributions but also ecosystem functioning [46]. These efforts allow the beginnings of the integration of large sets of biodiversity data with climatological parameters [50,51]. An important next key step is to incorporate into computational models perhaps the most pervasive of effects, the influence and effect of human activities such as production of global warming gases on climate. This is already under way and will be possible to model effectively by 2010.

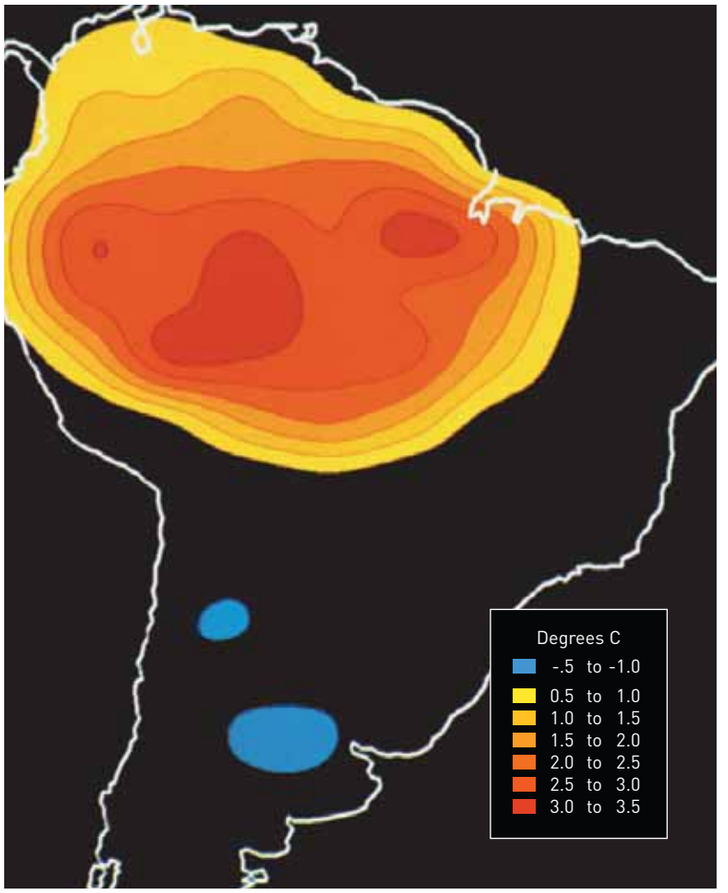
The increasing dependence on computing and computer science can be summarised in three key common trends discussed next.

Autonomous experimentation

Advances in remote intelligent sensor technology, coupled with advances in machine learning are expected by 2012 to enable both (i) autonomic observation (including identification of species [52]) and (ii) autonomous experimentation (see the subsection 'Artificial Scientists' in Part 2), providing highly comprehensive climatic, oceanographic and ecological data among others. It will, however, also hinge on effective distributed data management (see 'Semantics of Data' in Part 1). Some major efforts in this direction are the NEON (National Ecological Observatory Network; http://www.neoninc.org/) and the Long-Term Ecological Research Network (LTER, with its associated informatics programme; http://www.ecoinformatics.org/) initiatives. Within 10 years, these and other projects will enable access to a net of high resolution, real time ecological data.

Data management

The growth in, availability of, and need for a vast amount of highly heterogeneous data are accelerating rapidly according to disciplinary needs and interests. These represent different formats, resolutions, qualities and updating regimes. As a consequence, the key challenges are: (i) Evolution towards common or interoperable formats and mark-up protocols, e.g. as a result of efforts under way by organisations such as GBIF (www.gbif.org) or the recently created National Evolutionary Synthesis Center (www.nescent.org), we expect that by 2008 a common naming taxonomy incorporated in Web services will enable data from the diverse sources around the planet to be linked by any user; (ii) Capability to treat (manage, manipulate, analyse and visualise), already terabyte and soon to be



Effects of deforestation

Computer-generated map of South America, showing the mean surface air temperature change due to total replacement of rain forests with pasture. Temperature increases are colour-coded from pale yellow (0.5-1°C) to brown (3-3.5°C). Drop in mean temperature is shown as blue. The tropical rain forest reflects a very large proportion of infrared (heat) radiation that comes from the Sun. The loss of the forest would allow far more heat to be absorbed by the ground, thus raising local temperatures. This is one factor that would lead to a dramatically drier climate for the region.

Center for Ocean, Land & Atmosphere Interaction, Maryland / SCIENCE PHOTO LIBRARY

petabyte datasets, which will further increase dramatically when data acquisition by sensors becomes common. The developments required to support such activities are discussed extensively in Part 1 of this roadmap.

Analysis and modelling of complex systems

In ecology and evolutionary biology, analytical exploration of simple deterministic models has been dominant historically. However, many-species, non-linear, non-locally interacting, spatially-explicit dynamic modelling of large areas and at high resolution demands the development of new modelling techniques and associated parameter estimation and model validation [53]. Promising as these techniques are, by far the most challenging task is to integrate the heterogeneous, fast-growing amount of primary biodiversity data into a coherent theoretical (and hopefully predictive) framework. For example, the development of novel and efficient algorithms to link niche models used to predict species distributions to phylogenetic analysis in a spatially explicit context. Increased availability and use of data, and simulation power, without a comprehensive formal and theoretical scaffolding will not be enough. In addition to continuing developments in biostatistics and non-linear dynamics, computer science has the potential to provide theoretical paradigms and methods to represent formally the emerging biological knowledge. The development of formal languages oriented to represent and display complex sets of relations and describe interactions of heterogeneous sets of entities will help much of biology to become more rigorous and theoretical and less verbose and descriptive. (see the subsection 'Codification of Biology' in Part 2 and the section 'Global Epidemics' below).

Computer science and computing have an essential role to play in helping understand our environment and ecosystems. The challenges run from providing more powerful hardware and the software infrastructure for new tools and methodologies for the acquisition, management and analysis of enormously complex and voluminous data, to underpinning robust new theoretical paradigms. By conquering these challenges we will be in a much better position to manage and conserve the ecosystem underpinning our life-support systems.

Jorge Soberon, Stephen Emmott, Neil Ferguson, Tetsuya Sato

Understanding Biology

The Cell

Living cells are extremely well-organised autonomous systems, consisting of discrete interacting components. Key to understanding and modelling their behaviour is modelling their system organisation. Four distinct chemical toolkits (classes of macromolecules) have been characterised, each combinatorial in nature. Each toolkit consists of a small number of simple components that are assembled (polymerised) into complex structures that interact in rich ways. Each toolkit abstracts away from chemistry; it embodies an abstract machine with its own instruction set and its own peculiar interaction model. These interaction models are highly effective, but are not ones commonly used in computing: proteins stick together, genes have fixed output, membranes carry activity on their surfaces. Biologists have invented a number of notations attempting to describe these abstract machines and the processes they implement. Moving up from molecular biology, systems biology aims to understand how these interaction models work, separately and together.

Following the discovery of the structure of DNA, just over 50 years ago, molecular biologists have been unravelling the functioning of cellular components and networks. The amount of molecular-level knowledge accumulated so far is absolutely amazing, and yet we cannot say that we understand how a cell works, at least not to the extent of being able to easily modify or repair it. The process of understanding cellular components is far from finished, but it is becoming clear that simply obtaining a full part list will not tell us how a cell works. Rather, even for substructures that have been well characterised, there are significant difficulties in understanding how components interact as systems to produce the observed behaviours. Moreover, there are just too many components, and too few biologists, to analyse each component in depth in reasonable time. Similar problems also occur at each level of biological organisation above the cellular level.

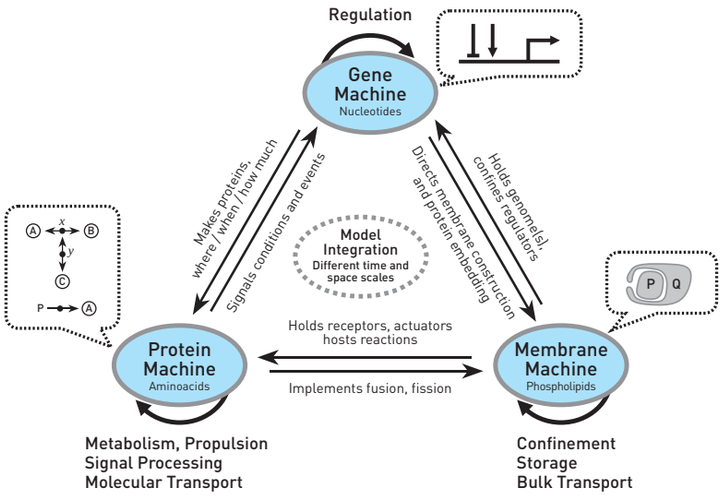
Enter systems biology, which has two aims. The first is to obtain massive amounts of information about whole biological systems, via high-throughput experiments that provide relatively shallow and noisy data. The Human Genome Project is a prototypical example: the knowledge it accumulated is highly valuable, and was obtained in an automated and relatively efficient way, but is just the beginning of understanding the human genome. Similar efforts are now under way in genomics (finding the collection of all genes, for many genomes), in transcriptomics (the collection of all actively transcribed genes), in proteomics (the collection of all proteins), and in metabolomics (the collection of all metabolites). Bioinformatics is the rapidly growing discipline tasked with collecting and analysing such omics data.

The other aim of systems biology is to build, with such data, a science of the principles of operation of biological systems, based on the interactions between components. Biological systems are obviously well-engineered: they are very complex and yet highly structured and robust. They have only one major engineering defect: they have not been designed, in any standard sense, and so are not laid out so as to be easily understood. It is not clear that any of the engineering principles of operations we are currently familiar with are fully applicable.

Understanding such principles will require an interdisciplinary effort, using ideas from physics, mathematics, and computing. These, then, are the promises of systems biology: it will teach us new principles of operation, likely applicable to other sciences, and it will leverage other sciences to teach us how cells work in an actionable way.

Many aspects of biological organisation are more akin to discrete hardware and software systems than to continuous systems, both in hierarchical complexity and in algorithmic-like information-driven behaviour. These aspects need to be reflected in the modelling approaches and in the notations used to describe such systems, in order to make sense of the rapidly accumulating experimental data.

We believe it is essential to look at the organisation of biological systems from an information science point of view. Cells are without doubt, in many respects, information processing devices. Without properly processing information from their environment, they soon die from lack of nutrients or from predation. We could say that cells are based on chemistry that also perform some information processing.



Abstract Machines of Systems Biology

An abstract machine is a fictional information-processing device that can, in principle, have a number of different physical realisations (mechanical, electronic, biological, or software). An abstract machine is characterised by a collection of discrete states, and by a collection of operations (or events) that cause discrete transitions between states, possibly concurrently. Biochemical toolkits in cellular biology (nucleotides, amino acids, and phospholipids) can be seen as abstract machines with appropriate sets of states and operations. Each abstract machine corresponds to a different kind of informal algorithmic notation that biologists have developed (inside bubbles). To understand the functioning of a cell, one must understand (at least) how the various machines interact. This involves considerable difficulties in modelling and simulations because of the drastic differences in the 'programming model' of each machine, in the time and size scales involved. Cardelli L., Trans. Comput. Syst. Biol. III, LNBI 3737, pp. 145 – 168. Reproduced with permission.

But we may take a more extreme position, namely that cells are chemistry *in the service* of information processing. Hence, we should look for information processing machinery within the cellular machinery, and we should try to understand the functioning of the cell in terms of information processing, instead of chemistry. In fact, we can readily find such information processing machinery in the chemical toolkits that we just described, and we can switch fairly smoothly from the classical description of cellular functioning in terms of classes of macromolecules, to a description based on abstract information-processing machines.

An *abstract machine* is a fictional information-processing device that can, in principle, have a number of different physical realisations (mechanical, electronic, biological, or even software). An abstract machine is characterised by:

- A collection of discrete states.
- A collection of operations (or events) that cause discrete transitions between states.

The evolution of states through transitions can in general happen concurrently.

Each of the chemical toolkits we have just described can be seen as a separate abstract machine with an appropriate set of states and operations. This abstract interpretation of chemistry is by definition fictional, and we must be aware of its limitations. However, we must also be aware of the limitations of *not* abstracting, because then we are in general limited to work at the lowest level of reality (quantum mechanics) without any hope of understanding higher principles of organisation. The abstract machines we consider are each grounded in a different chemical toolkit (nucleotides, amino acids, and phospholipids), and hence have some grounding in reality. Moreover, each abstract machine corresponds to a different kind of informal *algorithmic notation* that biologists have developed: this is further evidence that abstract principles of organisation are at work.

The *Gene Machine* (better known as Gene Regulatory Networks) performs information processing tasks within the cell. It regulates all other activities, including assembly and maintenance of the other machines, and the copying of itself. The *Protein Machine* (better known as Biochemical Networks) performs all mechanical and metabolic tasks, and also some signal processing. The *Membrane Machine* (better known as Transport Networks) separates different biochemical environments, and also operates dynamically to transport substances via complex, discrete, multi-step processes.

These three machines operate in concert and are highly interdependent. Genes instruct the production of proteins and membranes, and direct the embedding of proteins within membranes. Some proteins act as messengers between genes, and others perform various gating and signalling tasks when embedded in a membrane. Membranes confine cellular materials and bear proteins on their surfaces. In eukaryotes, membranes confine the genome, so that local conditions are suitable for regulation, and confine other reactions carried out by proteins in specialised vesicles.

To understand the functioning of a cell, one must also understand how the various machines interact. This involves considerable difficulties (e.g. in simulations) because of the drastic difference in time and size scales: proteins interact in tiny

fractions of a second, while gene interactions take minutes; proteins are large molecules, but are dwarfed by chromosomes, and membranes are larger still.

In the eyes of a computer scientist, this picture of cellular biology is quite remarkable. Not only is life based on digital coding of information (DNA), but it turns out that at least three of the biochemical toolkits used by living cells are *computationally complete*: they each support general computation, and they are each involved in some form of information processing. The implication is that many cellular processes should be considered as algorithms operating within some computational model. Many of these, both algorithms and models, are certainly still to be discovered.

Luca Cardelli

The Immune System

Protection of the individual from threats – whether originating from Earth or outer space – will be on top of any grand challenge agenda in the coming decades. During the evolution of life on the planet, the same agenda has indeed been in effect for billions of years. Organisms have developed complex defence set-ups known as immune systems fighting invading threats at the molecular level. The major assignment of an immune system is to defend the host against infections, a task which clearly is essential to any organism – and even to artificial systems such as computers and mobile phones. While surprisingly many other traits of the human organism can be linked to individual genes, immune systems have always been viewed as systems, in the sense that their genetic foundation is complicated and based on a multitude of genes and proteins operating in many different pathways, which interact with each other to coordinate the defence against infection.

Importantly, natural immune systems have been designed so as to differ for different individuals, such that any one type of pathogen or cancer most often will not be able to bring down the entire species. Human vaccines are normally designed to fight pathogens on a statistical basis, in the sense that they are not equally effective for all individuals in a population. In some cases, they even might be deadly to certain sub-groups. For most infectious agents, we do not have any preventive or therapeutic vaccine at all.

Understanding human immune systems by computational means represents a major challenge in the fight against emerging pathogens, cancers, inflammatory diseases, autoimmunity, and also in the transplantation area where knowledge about the permissible immune system mismatch between donor and recipient is essential. Rapidly changing infectious agents such as HIV and influenza, and even 'biowarfare' threats have added to the need for creating extremely fast, computational approaches for the design of new vaccines.

Virtual immune systems

Virtual human immune systems should be able to compute the result of host–pathogen interaction, including solutions to the pattern recognition problem of discriminating between self and non-self. They should be able to close the gap between experimental studies of the immune system and other human systems and the design of tools for clinical applications. The immune system includes an adaptive, learning system that has multiple levels (molecular, cellular, tissue, organ, organism, and organism-to-organism), and in this sense the task is 'worst-case' because it is not confined to any one single level. Still, computational approaches from the area of machine learning are well suited as frameworks for converting experimental data into algorithms capable of mimicking how this training takes place at each systemic level.

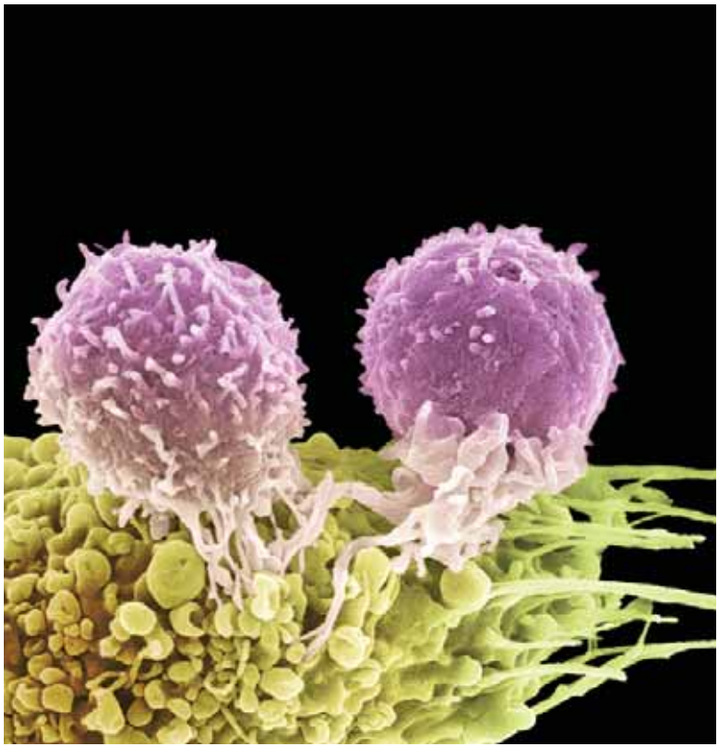
The immune system is a combinatorial system with a large number of 'computing agents' typically with $>10^{15}$ antibodies and $>10^{12}$ immune system cell clones in an individual. These numbers dwarf the 10^{6}-10^{7} non-immune-system products (including modified and alternative products) encoded by the human genome. It is therefore essential that the computational modelling takes into account the distributed and combinatorial nature of the immune system, which forms the basis for its ability to fight an astronomical number of pathogens and the ability to discriminate between self and non-self. Computational models of the immune system typically fall into three categories, involving molecular interactions (such as peptide binding to receptors), simulation of antigen processing (connecting several molecular level processing steps), and system-level interactions (such as models of overall immune response, or cellular models of the immune system). In terms of computing, these categories involve pattern recognition at the sequence analysis level (peptide binding and protein degradation), integration of such algorithms modelling different processing steps, and simulation of the huge, combinatorial aspect. The challenge is to bridge these different categories and to integrate them in ways that can compute a specific vaccine design tailored to individuals with different tissue types in the best possible way.

Systems biology, synthetic biology and theranostics

The study of human immune systems calls for a systems biology approach, where the computer will be essential for integration of exponentially growing amounts of data from genomics and proteomics, and for structuring formally systems descriptions such that they can form the basis for predictive simulation that can be used in the clinical setting and in regulatory agency approval procedures [54]. Within a decade, we believe advances in computational biology (see section 'Revolutionising Medicine' below) will make it likely that the complete DNA sequence for any individual will be determinable at very low cost, meaning that lists of parts of individual immune systems will also become more and more complete, leading to a much more realistic scenario for the new wave of large-scale computational analysis of such systems. This type of computational research will not only fall into the category of systems biology, but also represent 'synthetic biology' where the aim is – aided by computation – to build biological systems piece by piece. We wish to use computational approaches to build the protective capacity of immune systems in rational, non-reductionist ways, where the systems properties are taken into account and understood. Computational design of preventive and therapeutic vaccines also fits into the general area of theranostics, which describes the use of diagnostic testing to diagnose a disease,

choose the correct treatment regime and monitor the patient response to therapy. The aim is not only to describe by data integration the 'anatomy' of the immune system, but most importantly to understand its dynamics by simulation.

Søren Brunak



T lymphocytes and cancer cell

Coloured scanning electron micrograph (SEM) of two T lymphocyte cells attached to a cancer cell. T lymphocytes are a type of white blood cell and one of the components of the body's immune system. They recognise a specific site (antigen) on the surface of cancer cells or pathogens and bind to it. Some T lymphocytes then signal for other immune system cells to eliminate the cell. Cytotoxic T lymphocytes eliminate the cell themselves by releasing a protein that forms pores in the cell's membrane. The genetic changes that cause a cell to become cancerous lead to the presentation of tumour antigens on the cell's surface.

Steve Gschmeissner / SCIENCE PHOTO LIBRARY

The Brain

Understanding how the brain works remains a major challenge. Brains are plainly complex systems (containing 10^{11} neurons, and perhaps 10^{15} synapses), and we happen to know two interesting things about them that prompt caution about how much we will understand of how they work by 2020. One is derived by observation: brains drive almost all human behaviour, which is sometimes quite impressive, as in composition of symphonies, political systems, acts of generosity, science, roadmaps and so on. The other is derived from introspection: we know that it is *like something to be* (i.e. there is a subjectively experienced mental life associated with being) this particular type of system, in a way that we do not confront with any other system studied by science. These observations can induce defeatism, particularly in those outside neuroscience. Nonetheless, in another sense, the brain is simply an organ. Advances in understanding brain function both cause and follow a less numinous view of this particular organ system.

Neuroscientific understanding has made very substantial gains since the inception of neuroscience as a distinct programme in the 1950s and 60s. Then, cortical architecture appeared rather homogeneous; apart from a few gross fascicles, there was no evidence for connectionally structured systems; and neurophysiology suggested no more than simple local selectivity. There was little in this landscape that corresponded to sensory perception, or decision making, or memory; still less conscience, consciousness, or the qualia. However, neuroscience in the meantime has revealed a radically different landscape, in which canonical computational circuits are expressed in local cortical connectivity; beautifully complex large-scale connectional systems are present, with topologies that are both strongly suggestive of detailed function and also correlated in detail with physiological properties; and neuronal selectivity and computation is now known to bear the strongest association with perceptual and other functional states, and to be dependent on a very large amount of internal information about regularities and statistical properties of the world.

An explanation of how brains work will look much like an explanation of how any other system works. Its elements will be: (i) how the system is organised; (ii) what processes go on inside it; and (iii) how these processes interact to cause the behaviour of the system. It is also often helpful to know what the purpose of the system is. Every aspect of research on the key issues that represent elements of an explanation of how the brain works will require further experimental and significantly greater computational work.

The precise details of a processing system defined by 10^{15} synapses are vastly too great to be tractable, and it is possible that we will never be able to characterise fully how the brain is organised at the neuronal level. Fortunately, however, biology has provided many simplifying regularities. It has long been known that the brain is not a homogeneous mass in which neurons make connections willy-nilly, but is divided into many different areas, nuclei or processing compartments, in which neurons tend to have similar patterns of connectivity with distant brain regions. A further regularity is that the patterns of connectivity within most local

brain areas, particularly in the cortical structures, tend to be similar, and perhaps 'canonical'. These regularities have made possible significant progress, through characterising canonical patterns of local connectivity through microanatomy, and characterising the grosser areas and processing compartments, and their connectional relationships.

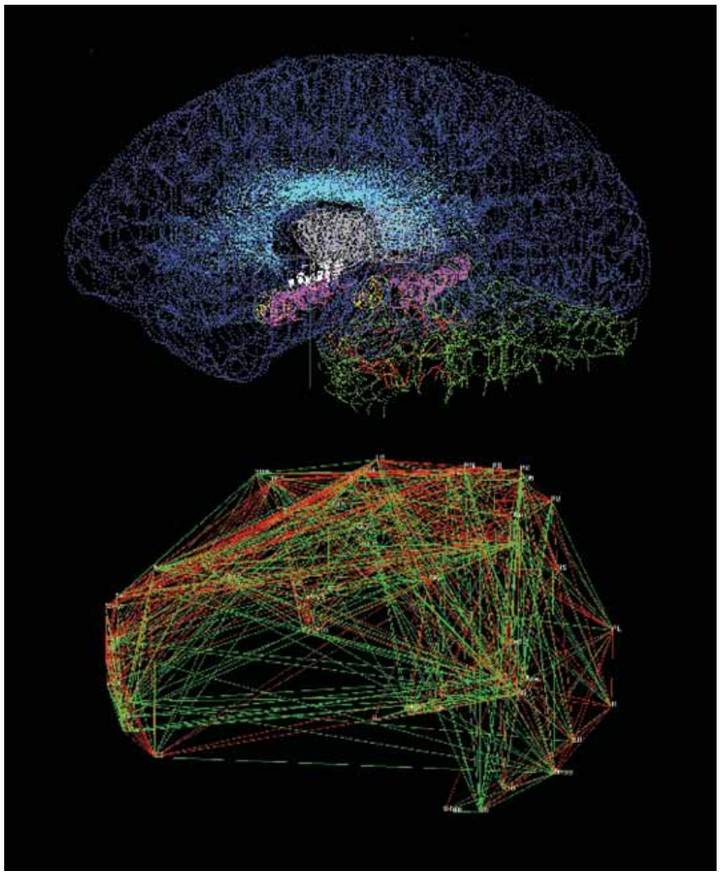
Computational work has already indicated that all central sensory systems are somewhat discrete, with each being ordered hierarchically (e.g. the lowest stations in each sensory hierarchy are the primary sensory areas, while the highest stations in each sensory system are typically connected to limbic and prefrontal structures). Experimental neuroanatomy of the kind most useful to refining these architectural features is regrettably not as popular as previously, but computational science increasingly holds promise for elaborating the architectures in which the brain's computations take place.

Understanding functional processes in brain systems is also advancing strongly. There will continue to be healthy differences of opinion about mechanisms of information processing in brain systems, but there is a (perhaps temporary) consensus among many in computational neurophysiology that cortical neurons see only spatio-temporal patterns of inputs, regularities in which they learn; that local computation is approximately Bayesian, where structural changes in connectivity from neuronal plasticity and deletion embed information learnt from experience in the network as indicants of prior probability; and that attention acts to change the gain between priors and afferents. Models with these properties do appear to account for very many observations of neuronal computation, but both experimental and computational research will be required to elaborate these processes.

The punch-line for understanding brain function will be delivered when an understanding of how complex brain processes interact to cause coherent behaviours becomes compelling. This will be the most difficult step, as studies *in vivo* are very challenging, and methods of recording brain activity in humans too indirect to constrain models definitively. We think it likely that this final step in understanding brain function will be made by advances in computational modelling, such as those based on Bayesian inference (see the section 'Prediction Machines' in Part 2) as they become increasingly capable of many of the more impressive behaviours evidenced by brains.

It is intriguing that almost all theoretical treatments of brain function suppose that *software is not meaningfully present* in brain computation: there is only dialogue between network structure and information. As computational models of brain and brain-like computing evolve towards 2020, it is likely that genuinely new computational artefacts will be generated, which will allow us to advance above the technology plateau represented by current computing machinery, and in turn enable greater capabilities for future computational science as it addresses future challenges.

Malcolm Young



A 3D model of the human brain (top)

The dark blue structures are the two hemispheres of the cerebral cortex, joined by a major fibre tract, the corpus callosum, shown in light blue. The cerebellum is outlined in green, the thalamus in grey, the hippocampus in pink, and the amygdala in yellow. The system rests atop the brainstem, pons and medulla, which are shown in red.

The topological organisation of the neural connections in the cat cerebral cortex (bottom)

The graph shows connections between each of the areas of the cortex, and has been computed to reflect the connectional topology of the systems in 3D.

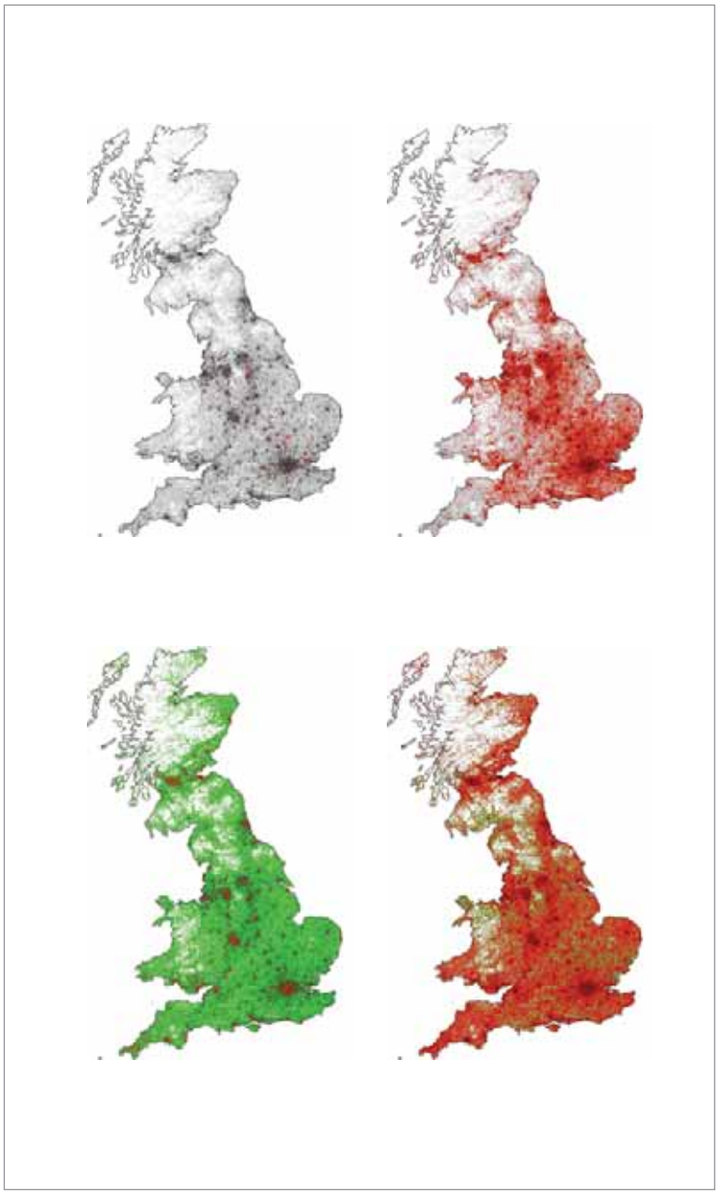
Global Epidemics

Recent years have seen animal and human populations challenged with a number of emerging and re-emerging infectious diseases – including H5N1 avian influenza, Nipah virus, Severe Acute Respiratory Syndrome (SARS), the 2001 UK foot-and-mouth disease (FMD) epidemic, Bovine Spongiform Encephalopathy (BSE)/variant Creutzfeldt-Jakob Disease (vCJD), West Nile virus and (by no means least) HIV, to name the best known. Nearly all these pathogens are zoonoses – diseases which originate in an animal host, but then cross species barriers – a factor which further complicates their evolution and epidemiology. The challenges facing humanity is to anticipate and prepare for new threats (whether natural or deliberate) where possible, and to identify and contain novel and unpredicted disease outbreaks as they arise. A particular concern is the potentially very rapid rate at which deliberately introduced or emerging infections might spread through our increasingly densely populated and interconnected world. For example, the large increases in the volume of passenger air travel over the past few decades undoubtedly played an important role in the rapid transmission of the aetiological agent of SARS between countries.

Computational science has a key role to play in confronting infectious disease threats in four key areas: (i) the rapid collection and analysis of high-throughput genomic and proteomic data in pathogen identification, diagnostic screening and molecular surveillance; (ii) enabling real-time epidemiological surveillance at both a regional and local scale; (iii) improving realistic predictive modelling of disease spread and the impact of control measures, both in preparedness planning and in real-time response to an unfolding outbreak; (iv) facilitating effective communication and management within rapidly formed ad-hoc global inter-disciplinary scientific and public health teams responding to a new outbreak. We focus on the third of these here – the first having much in common with the application of high throughput technologies in other areas of biomedicine (e.g. see the next section 'Revolutionising Medicine').

In planning for novel epidemics, it is important for contingency planning to rationally assess the possible scale of casualties that outbreaks of particular pathogens might produce, and to identify the set of control measures likely to be required to minimise the impact of such outbreaks. Given the potential speed of spread in some cases (e.g. a new influenza pandemic or another SARS like virus), advance planning is critical. Even slight delays or inefficiencies in responding can substantially increase the scale of an outbreak (as happened in the case of the UK FMD outbreak in 2001) – or potentially represent the difference between an outbreak being contained and control being lost. Assessing the optimality of interventions is complicated by the potentially negative consequences of some interventions: smallpox vaccination causes severe adverse effects in a significant minority of individuals, while interventions such as restrictions on movements between or within countries might have severe economic consequences (as well as human rights implications).

Mathematical models of the infectious disease transmission have been demonstrated to be valuable tools in outbreak planning and response, because they can integrate epidemiological and biological data to give quantitative insights into



Epidemic modelling

Simulation of the first 77 days of an avian H5N1 influenza epidemic in the United Kingdom (clockwise from top left at 13, 35, 56 and 77 days).

patterns of disease spread and the effect of interventions. However, meeting the challenge of emerging infections such as avian influenza requires response plans to be formulated at a range of scales: global, continental, national and local. A key question is what controls are most effective at each scale – and how actions taken at one scale or in one region (e.g. exit controls in the countries first affected) affect spread to and within other parts of the world. Currently, such questions tend to be addressed with models which are optimised for considering spread and control at one scale alone; for instance, we are now able to use individual-based simulations at a national scale (the largest currently under development are considering populations of 300 million), but when considering international spread, typically a much simpler 'patch' model is used.

However, as computational hardware, software tools (e.g. agent-based modelling languages and toolkits) and data availability improve, these limitations will be able to be overcome. A truly global simulator would capture population movements and contacts at all scales – from within each household to intercontinental travel – and could revolutionise our ability to understand, visualise and potentially predict patterns of spread of both novel and emerging pathogens and the impact of possible control measures.

The challenges in developing a useful and scientifically robust global simulator are significant, supercomputing hardware requirements being just one. Robust methods would also need to be developed for model validation and (potentially real-time) parameter estimation (using both historical data and data feeds from an unfolding outbreak). Optimally, population data would be updated in real-time (using transport system and mobile phone tracking data, for example), to enable the impact of population behaviour on disease spread to be assessed. However, once completed, such a simulation framework would represent a truly global resource for planning for and responding to future epidemics.

Neil Ferguson

Revolutionising Medicine

Understanding and eradicating disease present enormous scientific challenges, and, towards 2020, we expect that computing and computer science will play an increasingly vital role in addressing them. The eradication of disease implies the development of new therapeutic agents and regimens (we do not consider the vitally important issue of politics and economics here as it is outside the scope of this report), which in turn is critically dependent on the detailed understanding of disease. Clearly, these new therapeutic agents will have to be better and smarter than their predecessors.

Understanding disease

'Understanding disease' means that one can classify a disease state using phenotypic observations and molecular diagnostics, explain how the disease state is caused by abnormal molecular processes (e.g. modulation of enzymatic functions, binding affinities, etc.) and eventually explain the causes of these perturbations (e.g. polymorphisms or larger chromosomal abnormalities, pathogens, environmental factors). Although the sequence of the human genome – as well as that of model organisms and pathogens – is known, our knowledge of the function and physiological role of its individual genes and their transcripts has grown only modestly in comparison to the number of these genes. Consequently, our understanding of the molecular causes of disease hasn't progressed much either. This is largely due to the complexity of discovery sciences and the fact that genes do not operate in isolation but in regulated networks of variable size and complexity. We are only starting to understand some of the pathways within these networks and how their perturbations – induced by mutations or the unfavourable combination of certain gene variants and environmental conditions – are the causes of diseases. To understand disease, drug discovery can therefore no longer satisfy itself with reductionism and focusing on isolated genes as targets, but must take into account the full biological context of a target. Only in this way can the long-term productivity of drug discovery (as measured by 'new molecular entities' filed with regulatory agencies, relative to R&D spend) be improved. What might then constitute new, emergent approaches? High on the list of candidates is systems biology, which integrates experimental, mathematical and computational sciences in a heretofore unprecedented level of interdisciplinarity between medicine, biology, chemistry, physics, material sciences and information sciences.

Eradicating disease

Eradicating disease will require finding cures for diseases (including 'orphan diseases' which mostly affect economically challenged countries and 'small indications' which are perceived to hold only limited economical promise because of their limited demographics) that are clearly distinct from solely symptomatic relief. To create novel testable hypotheses, drug discovery will need to fully embrace systems biology, in such a way that chemistry also becomes part of the 'system' along with biology and medicine. This will enable the selection and validation of novel candidate drugs and treatment modalities including both small and biological molecules. Every drug or drug candidate will tend to interact with

more than one gene and therefore with possibly more than one pathway. This may cause undesired and toxic side effects. Drug targets can thus no longer be considered in isolation but must be studied in the context of the complex and dynamic network of biological components. Future drug discovery will thus need to take into account not only the context of drug targets participating in interacting pathways, but also many other aspects of biological context, such as expression patterns, molecular interactions (including complexes, channelling, transport, multi-scale interactions, and similar phenomena), compartmentalisation, structure, and phylogeny. This will allow for an improved understanding of the mechanisms of action of drugs, enabling (i) effective candidate optimisation to maximise therapeutic efficacy while minimising toxicity, and (ii) understanding their unexpected side effects to better exploit them for seemingly unrelated indications. Given these interdependences, it becomes apparent that understanding disease is a prerequisite to eradicating disease.

Integrative drug discovery

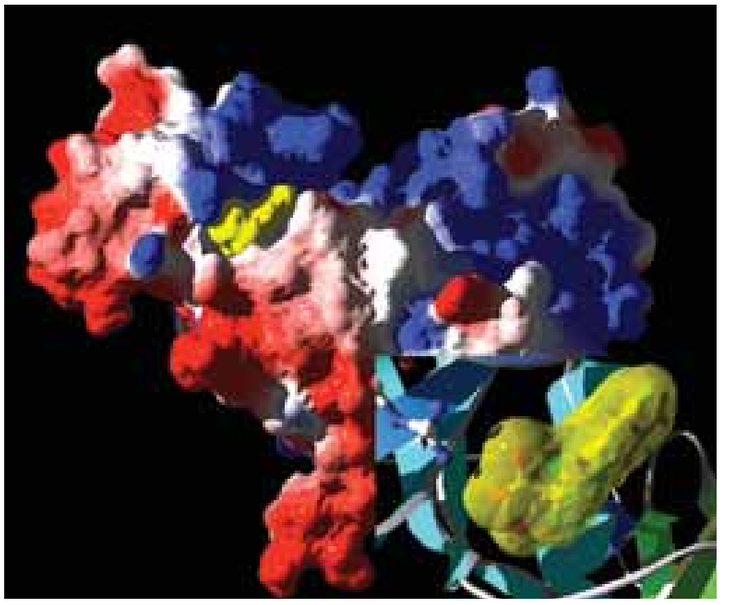
Truly effective data integration to support future drug discovery calls for more than architectural advances at the platform level, but new scientific approaches to the combination of data, and even cultural shifts as multiple disciplines are brought into apposition. Advances in computing and computer science outlined in Part 1 (together with the implementation of statistical and mathematical models) will play a pivotal role in understanding disease and selecting new drug candidates because of their crucial role in enabling systems biology, chemistry and biomedical engineering, and because they serve as a 'glue' between the scientific disciplines in drug discovery. Improved data mining and visualisation methods will also be required for better understanding of readouts based on profiles crossing many proteins, metabolites, and so forth, instead of single scalar values. Furthermore, bioinformatics, chemoinformatics and computational chemistry will mature and provide a collection of reliable methods, reflecting most of the key steps in drug discovery, integrated in an *in silico* drug discovery pipeline/workbench.

The contextual interpretation of scientific data and their transformation into knowledge and intellectual property lead to a wealth of scientific publications, patents and other reports. The scientific community is thus faced with a seemingly insurmountable amount of data and information, stored in a growing collection of databases, information sources and Web sites (see the section 'Transforming Scientific Communication' in Part 1). A major consequence of this complexity is the increasing need for effective 'computational knowledge management'. Consequently, much effort will be going into the integration of drug discovery and public 'omics' data and information, the development of information mining and extraction methods (applicable to text and images), natural language processing and the navigation across information domains at the conceptual and semantic level. These advances will be supported by key developments in text-based biological and chemical entity recognition methods, nomenclatures and knowledge representations (ontologies and nomenclature), machine learning and computer-based reasoning.

Smart drugs and molecular computers

The aforementioned paradigm shift to a computationally-driven systems biology approach will not only shorten drug discovery time, but hopefully drive the development of 'smart drugs'. 'Smart drugs' is a convenient rubric under which to discuss two important futures for drug discovery: (i) smarter ways to develop drugs that cure rather than only relieve symptoms, and that are safe, effective, and affordable; (ii) drugs that are themselves smart in some sense, e.g. targeted, adaptive, and so forth.

Drugs themselves may be 'smart' in several senses. It is increasingly clear that the simplistic view 'one drug fits all' is being superseded by a recognition of variations in responses within human populations, either in terms of efficacy or susceptibility to adverse reactions. Where such variation has a genetic basis, pharmacogenetics will be key to both the development and employment of new drugs. Rapid genotyping of individuals will allow for more and more customised therapies, with less prescribing trial-and-error and greater safety. New molecular biomarkers



Drug/target-interaction modelling

A potent and selective protein kinase CK2 inhibitor (in yellow in main picture and inset) identified by high-throughput docking of a large compound library into the ATP-binding site of a homology model of the CK2 kinase [55].

will allow for closer monitoring of drug effects in certain conditions. Combination therapies will increase the effectiveness of drugs by simultaneously addressing multiple causal axes or through mutual reinforcement.

Simple drugs disperse throughout the body and operate immediately and everywhere. Much effort is being invested in drugs that target a specific organ, tissue, or even a cell, and operate only conditionally. Ultimately, a smart drug would become active only upon reaching the correct destination and determining that a complex set of disease-related conditions is present at its destination. The resulting versatility in the administration of drug regimes will be matched by increasingly sophisticated drug delivery technologies, possibly to include increasingly 'smart' nanotechnologies, transdermal delivery devices, and other indwelling mechanisms, all of which may include sensor-based control or feedback schemas. These will require advances in biomedical engineering and much closer interaction than heretofore between the pharmaceutical and medical devices industries.

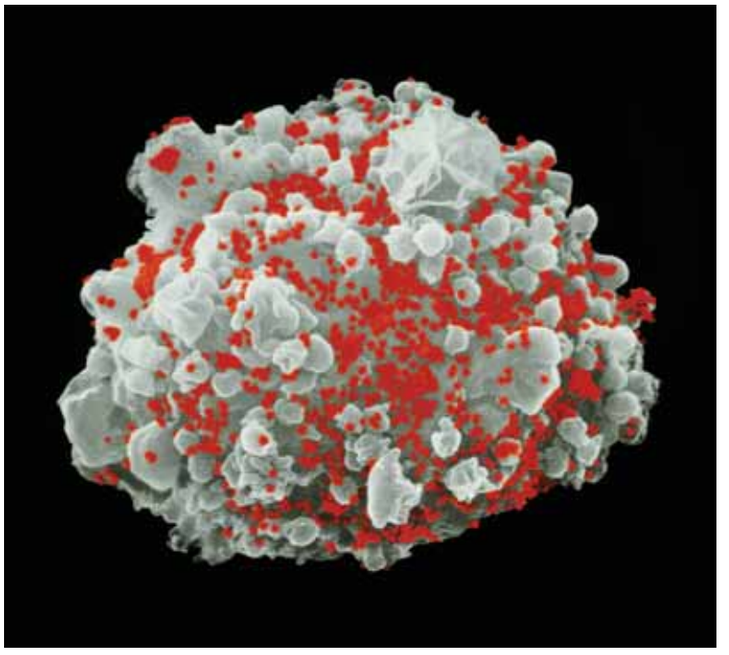
In the longer term, computing capabilities embodied in biomolecular systems may enable ever more sophisticated diagnosis, and provide for much stricter and refined preconditions for the in situ release of highly potent drug molecules, molecules that may be too harmful to administer without such sophisticated control.

Recent research [37] has demonstrated preliminary proof-of-concept of such a form of smart drug: a molecular computer that, at least *in vitro*, senses disease-related molecular symptoms, namely over – and under-expressed genes, analyses the information, and if it diagnoses a disease, releases the drug molecule it carries. Biologists may ask, "why call this a molecular computer when it is just a collection of molecules that undergo reactions according to the laws of chemistry?", but by the same token, is an electronic computer just a collection of wires and transistors that conduct electrons according to the laws of physics?

Importantly, this molecular computer-as-smart drug utilises several key computer science concepts: information, computation, input/output relation, propositional logic, stochastic automata, modularity, interfaces, parallel computation, specification versus implementation, interchangeability of program and data, interpreters, reliability, and building reliable systems from unreliable components. While such a molecular computer faces many hurdles before it could be of medical value, we predict that, ultimately, constructing a 'smart drug' that can be used medically will require using fundamental concepts from computer science.

Whilst 2020 science will drive a revolution in drug discovery, the development of smart drugs in this sense is emblematic of a critical aspect of future drug discovery: the societal and ethical questions that will be raised by new capabilities for intervention. Where does one draw the line between restorative therapy and enhancement of natural states or abilities? To what extent can we, or should we, rise above our genetic heritage? Who should decide about the availability of such therapies, and what limits should be imposed? For these difficult questions, science, alas, is unlikely to help.

Manuel C. Peitsch, David B. Searls, Ehud Shapiro, Neil Ferguson



A T-lymphocyte white blood cell infected with the HIV virus

HIV (Human Immunodeficiency Virus) is the cause of AIDS (Acquired Immune Deficiency Syndrome). This T-cell is from a culture cell line known as H9. An infected T-cell typically has a lumpy appearance with irregular rounded surface protrusions. Small spherical virus particles visible on the surface (red) are in the process of budding from the cell membrane. Depletion in the blood of T4 lymphocytes through HIV infection is the main reason for the destruction of the immune system in AIDS.

NIBSC / SCIENCE PHOTO LIBRARY

Understanding the Universe

Understanding the origin, workings and ultimate fate of the Universe is one of the great questions which has always fascinated mankind. This generation has the audacity to believe that it may be possible to finally answer these questions, but in order to do so, computational tools will need to be brought to bear on an unprecedented scale. The challenges to be faced are as difficult, but in many ways complementary, to those facing biologists in their quest for the origin of life.

Recent data from the COBE and WMAP [56] satellites, which measured temperature fluctuations of the Cosmic Microwave Background (CMB) generated in the early Universe, has moved cosmology into the realm of precision science. The data are in agreement with the consensus hot Big Bang model (known as Λ-CDM), with the Universe being created 13.7±0.2 gigayears ago. However, the data indicate that the Universe is made of only 4% ordinary matter, 23% of an unknown form of dark matter, and 73% a mysterious dark energy. The birth of the Universe occurred in a tiny quantum fluctuation which was driven by energy stored in the vacuum during a period of 'cosmic inflation', to reach the size observed today. It is therefore now necessary to investigate the nature of dark matter and dark energy, and to understand the properties of the quantum vacuum itself, especially in the unexplored regime of very high gravitational field which would have occurred during the Bang. More information can be gleaned from astronomical observations, of the CMB itself, of large-scale surveys of the structure of galaxies and galaxy clusters, and of high gravitational fields near massive objects. However, optical observations are limited to times later than about 380,000 years after the Bang, when the CMB was formed. To obtain direct evidence of the nature of dark matter and dark energy, and to examine conditions at earlier times, particle physics experiments are needed.

Particle physics has its own Standard Model (SM), which successfully describes all the observations so far. Matter is built from quarks and leptons, interacting through forces whose properties are set by fundamental symmetries of nature (gauge interactions). The model unifies the electromagnetic and weak nuclear forces into a single electroweak interaction, whose properties have been verified to the 0.1% level. This is a breakthrough as significant as the unification of electricity and magnetism by Maxwell. The model predicts the existence of the Higgs boson, which is responsible for particle masses in the model. The Higgs would be a completely new type of field, in the same class as is needed to drive inflation in the Big Bang. It is known that the SM needs extension, and possibilities include new particles which could form dark matter, and even extra space dimensions, giving the possibility of multiple universes in a higher geometry. Particle experiments can simulate the conditions at 10^{-20} seconds after the Bang.

Experiments in these fields are totally reliant on advanced computation. Indeed, particle physics experiments already reject 99.999% of all collisions before writing them to mass storage, using pattern recognition algorithms running in real time on the detectors. The next generation will require computational tools on an unprecedented scale. The particle physics problem is conceptually simpler than astronomy or biology, since the data acquired at accelerators can be handled in a

uniform manner. The Large Hadron Collider, which will start operations in 2007, will generate several petabytes of data each year, with the data stored and processed on a worldwide federation of national grids linking 100,000 CPUs (see the section 'Computational Science' in Part 1).

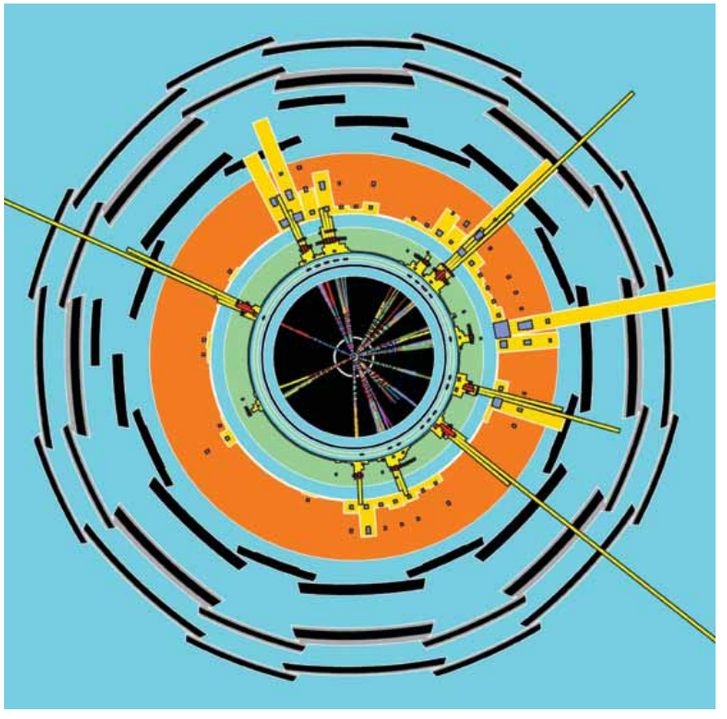
More challenging again will be to supply tools which enable the thousands of end users to analyse the physics objects. Here usability will be a key issue, since all aspects of data cataloguing and code handling will by necessity be automated: the user cannot expect to know the location of the data required, nor the availability of resources to process it. A completely transparent interface between the desktop analysis GUI and the computational resources will be needed to enable the user to focus on the science rather than the infrastructure to perform it. The GUI itself will have to run user defined analysis algorithms within a framework allowing the use of high level scripting languages combined with compiled code libraries. It will also require a move away from batch processing, as in present-day Grid implementations, to a much more interactive, though distributed, method of working. This will require data storage solutions combining the features of relational databases and conventional filestores, with advanced caching features across the Wide Area Network (WAN) to avoid intolerable latency issues.

In astronomy, the large-scale surveys of the structure of the Universe are creating data sets of similar scale, up to 5 PB/year, but without the same uniformity of data standards used in particle physics. The Virtual Observatory project aims to federate the data from different astronomical instruments, allowing users to access all the information about astrophysical objects simultaneously, thereby allowing a new form of virtual astronomy to take place, searching for example for variable sources, or new objects classified as outliers in smaller samples. The system can also be used in real-time to detect new sources like gamma ray bursts, and use robot telescopes to observe them before they fade away. This requires the ability to perform rapid data-mining across heterogeneous datasets, visualisation across the network, and automated pattern recognition. Here input from other computational sciences can be vital, with neural networks and machine learning algorithms applied to the data. Mining data in published literature is an important component, with opportunities to apply natural language processing techniques.

Understanding the large-scale structure of the Universe requires that the data from surveys are compared to models which predict the distribution of galaxies. This requires massive high performance computing (HPC) resources, such as those deployed by the Virgo consortium [57]. A single simulation of a 2 billion light year cube tracked 10 billion particles, and required a month's running, producing 25 terabytes of stored data. Ideally, such runs would be performed for many possible model parameters (and rival models) to compare to the data. Similarly, in particle physics, models such as supersymmetry can have over 100 free parameters. Both fields are adopting Bayesian methods of inference for such problems in order to search large parameter spaces efficiently, and to make robust statistical inferences. There is a great potential for common tools for computation steering, model fitting and data analysis.

The next decade will see the big sciences move to an era of massive federated data sets, and correspondingly massive data processing. In order to enable science to be extracted from this data deluge, it will be necessary to employ new methods of data federation, distributed computation, visualisation and interactive analysis. Cutting edge computer science solutions will need to be brought to mainstream applications for this to happen. In areas such as biology, the issues with handling and mining heterogeneous data are similar, and the scale of data is growing. When attempts are made to simulate whole cells, organs and organisms, the modelling and inference problems will also reach a similar scale. The need for a common effort on scientific computation to enable new advances is therefore evident.

M. Andy Parker



ATLAS simulation

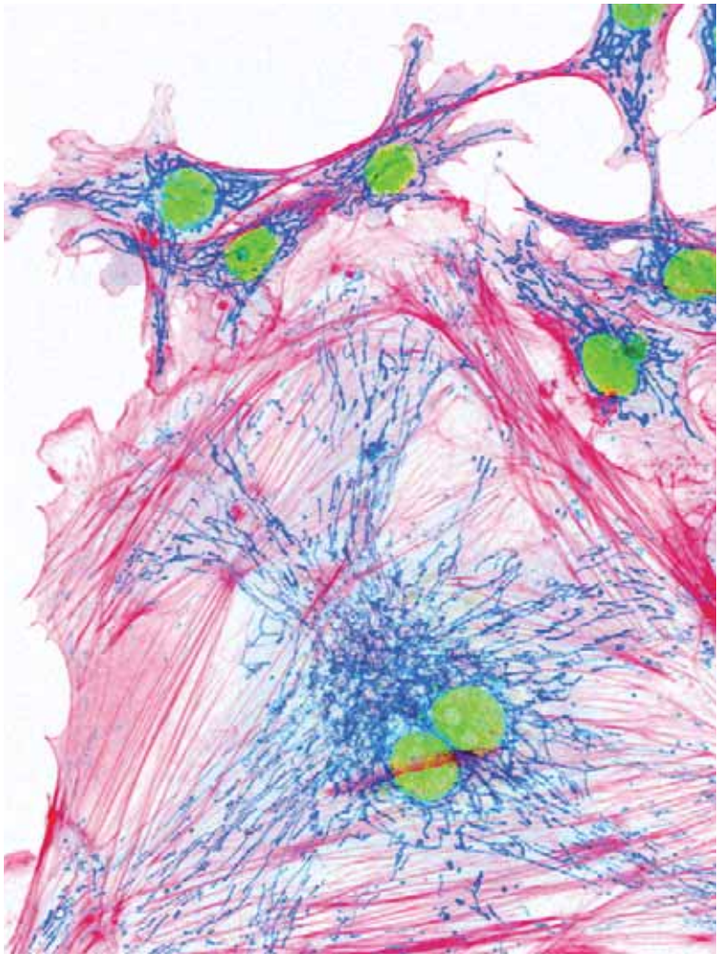
A simulation of a quantum black hole evaporating in the Large Hadron Collider's ATLAS detector.

The Origin of Life

‘What is the origin of life?’ This perhaps the most fundamental open question in science. While the last century brought a deep understanding of the molecular basis of life, very little is known about the mechanisms that afforded the emergence of life. Many biologists and biochemists believe that the problem lies in the realms of the chemistry of early Earth. However, the alternative ‘panspermia’ theory, originally proposed by Hermann von Helmholtz in 1879 and later embellished by Francis Crick [58], suggests an extra-terrestrial origin of life and therefore renders speculations on the specific chemistry of early Earth to be less fundamental to the question. The theory that life originated somewhere else may seem circular, as it seems to merely defer the question of the origin of life to some other planet. However, it does solve what seems to be a major paradox. Theories on the origin of life agree that at some point there was a ‘precursor’ cell that gave rise to all cell types and organisms as we know them today.

That cell had at least DNA, RNA, proteins, membranes, ribosomes and polymerases. The estimate is that such cells existed as early as 3.6 billion years ago, which is less than 0.5 billion years since the Earth cooled. Looking at the intricate cross relations between these components of the primary cell, and the fact that none seem to be able to exist without the others, the evolutionary distance between the primary cell and modern humans seems smaller than the distance between innate matter and the primary cell. Hence more time might have been needed to allow for its evolution than is afforded by planet Earth, which is precisely what Crick’s theory provides. Once we open the possibility of life originating somewhere else, we also have the freedom to speculate what conditions might have been most conducive to life’s development, without being confined to the specific conditions prevalent in early planet Earth. Even with this leeway, the challenge for origin of life researchers remains formidable: to demonstrate, *in vitro*, one or more evolvable self-replicating systems that could serve as feasible hypothetical milestones between innate matter and the primary cell. As a result of the improbability of reproducing the evolutionary process resulting in a primary cell in a lab within our scientific lifetime, computer analysis, modelling and simulation are essential to suggest such intermediate evolutionary milestones.

Ehud Shapiro



Confocal light micrograph of cultured endothelial cells

A fluorescent dye has been used to show the cell structure. Nuclei (green) contain the cell's genetic information, DNA (deoxyribonucleic acid) packaged in chromosomes. Actin filaments, part of the cell cytoskeleton, are pink. The cytoskeleton is responsible for the structure and motility of the cell. Actin is the most abundant cellular protein. The golgi apparatus, or body (blue), is a membrane-bound organelle that modifies and packages proteins. It consists of long flattened vesicles arranged in stacks.

David Becker / SCIENCE PHOTO LIBRARY

Future Energy

The need for new approaches to energy research

Secure and affordable energy supplies are vital for economic and social development. Existing drivers and by-products of economic and social development increase demand for energy, most noticeably in countries such as China and India that are rapidly making the transition to economic and industrial prominence. As a result, world demand for energy is rising steadily. Known reserves of conventional carbon-bearing energy sources, largely oil and gas, are being extended with new discoveries, but they are generally in geologically and geopolitically more difficult areas. Increases in annual production to meet demand will outstrip extensions to reserves and reserve-to-production ratios will fall, in some cases to single figures, in the most industrialised countries. Reserves of coal may last much longer as a future electricity-generating fuel, for instance in excess of 200 years in USA, but coal emits more CO_2 than oil or gas as do heavy-oil or oil shale fuels. Advances in exploration techniques and further exploitation of mature oil and gas fields with enhanced oil recovery technologies may increase the known carbon-bearing reserves and security of these fuel sources. However, even if these additional reserves were adequately to increase security of supplies, their combustion would add to the already rising CO_2 volume in the atmosphere unless effective and economical carbon sequestration techniques were to be applied on a strategic scale.

Climate change and its effects may be partly attributed to increased atmospheric CO_2 concentrations and the Kyoto protocol seeks to reduce annual production of CO_2. As a group, the EU agreed to reduce its emissions by 8% below 1990 levels for the first Kyoto commitment period between 2008 and 2012. Within this, the UK must reduce its CO_2 emissions by 12.5% and has set a domestic ambition of achieving 20% reductions by 2010 and 60% by 2050. Reduction of the CO_2 component of the basket of emissions requires the volume of carbon in the energy supply chain to be lowered, primarily in the transport and electricity production sectors. Demand reduction is most effective since energy ‘not-used’ reduces carbon flows all the way down the energy supply chain and conserves the hydrocarbon resource. There may be limited expectations of demand reduction in the short term since this is perceived to restrict industrial growth, social development and personal lifestyles. However, some demand reduction could be enabled by: improving fuel use and efficiency; heat and electricity demand management; greater deployment of domestic micro-generation and industrial combined heat and power plants. Successful operation and monitoring of such an evolved end-usage of energy will depend on advances in widely distributed data capture, processing and communication. Enabling economic and social growth and also meeting future demand for energy in an environmentally and economically sustainable manner present significant global and local challenges. Demand must be reduced where possible, existing and new resources of conventional fuels must be used more prudently, and alternative sources of energy must be developed into commercial viability.

A Systems Approach to Future Energy Research

Given the need to conserve carbon-bearing energy sources for as long as possible, it is imperative to find and advance new sources of low-carbon or renewable energy, including biomass, marine, photovoltaic, fuel cells, new-fission and fusion and to develop carbon management technologies. This will require fundamental and applied whole-systems research across the engineering, physical, life and social sciences. The energy system is perhaps one of the largest, naturally-, socially-, and capital-intense examples of an interacting system. It may only be developed by whole-system research spanning disciplinary boundaries. At the heart of this interdisciplinary research over the next two decades there will need to be new kinds of conceptual and technological tools operated on the multiple intersections of, for instance, biology, engineering, geology, communications, meteorology and climatology. There will need to be significant and enabling scientific advances in new materials, computing, data capture and synthesis, communications, modelling, visualisation and control. Highly novel approaches will need to be applied in research activity all the way down the energy supply chain to help establish viable future energy sources.

For example, spatial and temporal characterisation of the main marine renewable energy resources – offshore-wind, wave and tidal-currents will require advances in atmospheric sensing, climate and weather data acquisition in order to be able to model fully the behaviour and interaction of wind, sea and devices down to and through the air-water boundary. This will increase the ability to predict the variable energy production, forecast and survive storms and improve coastal defence. Real-time modelling of the combined effects of waves and tidal-currents will also be necessary to predict device interaction, reliability and survivability. This will not only require next generation High Power Computing capabilities but will also require powerful new predictive modelling capabilities to model the combination of meteorological, climatic, oceanographic, environmental and social effects, against a backdrop of climate change. This type of modelling capability may be made possible by the kinds of advanced modelling approaches outlined above in the 'Prediction Machines' section.

Bio-energy crops, as another example, offer large untapped potential as a source of renewable energy with near-carbon neutrality if grown, harvested and converted efficiently and with predictable performance. Production of both solid fuels (combusted dry and wet biomass) and liquid fuels (bioethanol, biodiesel) from a wide range of dedicated energy crops (such as grasses and short rotation trees) and food crop plant sources (including wheat, maize and sugar beet) in an environmentally sustainable manner requires better synergies between fundamental biological discoveries in genomics and post-genomics, and the application of massive computing power. This 'systems biology' approach aims to harness the information from molecules through to whole organisms - from DNA to transcriptome and proteome and beyond. Technologies in high throughput biology will generate vast data in this area over the coming decades and advanced computing processes and technologies will be vital, from the development of grid systems for data sharing through to producing new algorithms to make predictive models for enhanced plant performance and 'designed' plant quality. Similar principles can be applied to harnessing the power of micro-organisms where enzyme systems for degrading ligno-cellulose may be available but not yet applied in commercial systems. Biohydrogen and artificial photosynthesis provide another biological resource for future development and identifying natural variation and mutations in DNA that may be relevant to evolving these technologies will again rely on development of new computing approaches. At the other end of this energy chain there are coal-fired power stations that could be readily adapted for co-firing with biomaterials such as straw and coppice wood. However, optimised co-firing would introduce new challenges in design and prediction. Modelling flow and combustion for particulates with such disparate densities, sizes and compositions as coal and straw requires an improvement in complexity, resolution and visualisation of the flow, combustion and energy processes amounting to several orders of magnitude over that which is currently being used.

These are just two examples of the many renewable energy resources that could become part of a lower-carbon electricity supply by 2020, but integration of renewable resources with the electricity network in most countries presents another barrier that must be removed. Many of these resources are most abundant in the least densely populated areas, where the electricity distribution network was originally installed to supply increasingly remote areas of lower demand. The network there is largely passive and not actively managed with outward uni-directional power flows from the centrally dispatched power plants connected to the transmission network. The number of generators connected to the future distribution network could be orders of magnitude greater than the number of larger plants currently connected to the existing transmission network. Power flows will reverse in many areas and will be stochastic in nature. Rapid collection, transmission and processing of data to produce near real-time control responses to this stochastic change will be a key to successful operation of such a future electricity supply system. Assimilation of the data and state estimation of the network to anticipate and control changes in the system will require bi-directional communication. The volume and speed of data flow and its processing will need to be increased by several orders of magnitude. There will need to be new data aggregation, storage, and processing algorithms to enable the new control techniques at the ends of the two-way communication. Machine learning and near real-time optimisation techniques may offer this advance in the period up to 2020.

There are many advances that will be necessary to realise these and other future sources of energy, and to understand and mitigate change to the natural environment due to the renewable energy conversion. This will have to be set in the whole-systems interdisciplinary context and calls for computing and computer science roadmaps to plan the journey for the techniques and technology to the end points that support this vision. To this end, this 2020 science roadmap serves a vital purpose alongside the other research road-mapping taking place in the individual technologies.

A. Robin Wallace

Industrial air pollution

Steam and smoke clouds rising from the cooling towers and smokestacks of a chemical factory. Photographed in Grangemouth, Scotland.

Jeremy Walker / SCIENCE PHOTO LIBRARY

Postscript: Building Blocks of a Computing Revolution

Discussion thus far has been largely about the role of computer science and computing in transforming, even revolutionising science. There also exists the distinct possibility that the opposite could occur as a consequence: that such advances in science, especially biology and chemistry, could create the building blocks of a fundamental revolution in computing.

Computers as we know them excel at the tasks they were conceived for. Yet, increasingly, one can envision applications of information processing for which the established computing technology is unsuitable. Bioimmersive computing devices that would operate within a living organism, or even inside a living cell, are an example [37]. Their realisation requires a complete information processing architecture smaller than a single transistor. No fundamental limit stands in the way of such a technology, as is amply demonstrated by the sophisticated intracellular information processing found in organisms. Similarly, in the area of robotics, real-time processing of complex data streams in a low-power, tiny, lightweight unit is at present out of reach – yet a broad range of social insects (such as ants) illustrate what would be possible by such robots given an appropriate technology.

Information processing is essential for biological systems, both to maintain their intricate organisation and to compete with rival life forms. Consequently, even the simplest organisms evolved enviable capabilities to tackle computationally difficult challenges. The principles of natural information processing methods are yet to be fully understood, but progress in the biosciences continually unveils more detail. What is known already informs the development of molecular computing concepts [59].

Today's computers have been designed to follow strictly a formalism imposed independent of their physical implementation. The properties of the materials that implement the computation are hidden by careful engineering. Albeit convenient for programming, this is an inefficient use of the computing substrate resulting in relatively large computing devices which are based on vast networks of identical, fast and simple switches. In contrast, the course of computation in nature's molecular computers is directly driven by the physicochemical properties of the materials that implement the computation.

The unique properties of macromolecules in particular afford the possibility of highly integrated information processors. Macromolecules are large enough to possess specific shapes, yet small enough to explore each other by diffusion. Short-range additive forces allow them to overcome entropy at relatively high temperature to self-assemble in selective combinations. Another important property of macromolecules is their conformational dynamics, i.e. their ability to change shape, and as a consequence function, in response to their local physicochemical environment. Through these largely stochastic processes, macromolecules provide much more powerful components than conventional silicon architectures [60]. The combinatorial building block principle for assembling specialised macromolecules offers an inexhaustible set of basic functions.

Two substantial challenges need to be overcome for exploiting the potential of a molecular information technology. First, computational concepts tailored to the

physics of macromolecules need to be worked out. Second, methods to orchestrate the structure and interaction of ensembles of molecules have to be developed. The magnitude of these challenges, however, is matched by the opportunities that will be laid open through the resulting technology. Exploratory research into both aspects is well under way. Indeed, a first step towards an 'intelligent drug' that would determine the disease state of a cell from within and act accordingly has already been tested in the laboratory. Ongoing research along this line may eventually lead to the integration of artificial control structures into cells with broad potential for applications in medicine and environmental sensing.

The change of computing substrate may also necessitate a change in the underlying model of computation. The von Neumann architecture, which underlies all past and present programmable electronic computers, is not relevant to computers made of biomolecules. Fortunately, the plethora of abstract computing devices explored within theoretical computer science has ample models to choose from. Specifically, the Turing machine, which stands at the foundation of theoretical computer science, has many similarities to molecular machines of the living cell: its unbounded tape resembles information encoding molecules such as DNA, RNA and even proteins, and its local processing operation resembles polymerases and the ribosome much more than an electronic computer. Indeed, it has served as inspiration for many theoretical and experimental molecular computing systems.

The new computer revolution will complement and extend, not overthrow, established methods. Enabling a wide range of novel uses for information processing, it will lead to materials, components and devices capable of responding with life-like adaptability to their environment.

Klaus-Peter Zauner, Ehud Shapiro

4 Conclusions and Recommendations

Conclusions

From our analysis and findings, we draw three conclusions about science towards 2020:

First, a new revolution is just beginning in science. The building blocks of this revolution are concepts, tools and theorems in computer science which are being transformed into revolutionary new conceptual and technological tools with wide-ranging applications in the sciences, especially sciences investigating complex systems, most notably the natural sciences and in particular the biological sciences. Some of us argue that this represents nothing less than the emergence of 'new kinds' of science.

Second, that this is a starting point for fundamental advances in biology, biotechnology, medicine, and understanding the life-support systems of the Earth upon which the planet's biota, including our own species, depends. In other words, that the scientific innovation already taking place at the intersection of computer science and other sciences ranging from molecular biology, organic, physical and artificial chemistry and neuroscience to earth sciences, ecosystems science and astrobiology has profound implications for society and for life on Earth. Additionally, such advances may also have significant economic implications. The new conceptual and technological tools we outline here have the potential to accelerate a new era of 'science-based innovation' and a consequent new wave of economic growth that could eclipse the last 50 years of 'technology-based innovation' characterising the 'IT revolution'. Economic growth from new health, medical, energy, environmental management, computing and engineering sectors, some of which are unimaginable today is not only entirely plausible, it is happening already. It is occurring as a consequence of the first stages of the scientific revolution now under way, a good example of which is the mapping of the human genome and the technological and economic innovation that has emerged from it.

Third, the importance and potentially profound impact of what is occurring already at the intersection of computing, computer science and the other sciences – the basics of which we summarise in this report – is such that we simply cannot afford to ignore or dismiss it. We need to act upon it. It is worth restating that our efforts have not been that of 'forecasting' or 'predicting'. We have simply summarised the developments actually occurring *now*, together with what we *expect* to occur as a consequence of emerging advances in computing and science, and what *needs* to occur in order to address the global challenges and opportunities we are already presented with as we move towards 2020. Government leaders, the science community and policy makers cannot afford to simply 'wait and see' or just continue 'business as usual'.

We are in important, exciting, indeed potentially extreme, times in terms of the future of our planet, our society and our economies, and extreme times call for bold measures. We therefore recommend the following immediate next steps as a call to action for the science community, for policy makers, and for government leaders.

Recommendations

1 Establish science and science-based innovation at the top of the political agenda

Politicians like to claim that science is important and vital to the future of the economy. They now need to back this claim with action and put science in the premier league of the political agenda. In a way we have never seen before, science really will be absolutely vital to societies, economies and our future on this planet towards 2020; and science-based innovation is likely to at least equal technology-based innovation in its contribution to economic prosperity. Making sure this happens will require governments to be bold about science and its role in the economy and society.

2 Urgently re-think how we educate tomorrow's scientists

Education policy makers need urgently to re-consider what needs to be done to produce the kinds of scientists we shall need in the next decade and beyond. Tomorrow's scientists will be amongst the most valuable assets that any nation will have. What is clear is that science will need new kinds of scientists, many of whom will need to be first-rate in more than one field of science as scientific research increasingly needs to occur across traditional scientific boundaries. As well as being required to be scientifically and mathematically literate, tomorrow's scientists will also need to be computationally literate. Achieving this urgently requires a re-think of education policies now, not just at the undergraduate and postgraduate training level, but also at the school level since today's children are tomorrow's scientists. The education of today's children – tomorrow's scientists – is something of such importance that no government can afford to get wrong, for failure to produce first-rate intellectual capital in a highly competitive emerging era of 'science-based innovation' will almost certainly carry with it serious economic consequences. Some specific recommendations are:

Children: (i) Take far bolder measures to interest children in science and then retain their interest in it and its importance for society; (ii) urgently and dramatically improve the teaching of mathematics and science in schools; (iii) make teaching of computing more than just 'IT' classes and how to use PowerPoint®. Make basic *principles* of computer science, such as abstraction and codification, a core part of the science curriculum.

Undergraduates: (i) Make computer science (again, not just 'computing') a key element of the science curriculum; (ii) develop into undergraduate education the concept of 'computational thinking' (see section on this topic in Part 1).

PhD students: (i) Training in research methods (experimental, mathematical and statistical methods) needs to be broadened to include computational methods; (ii) because of increasing interdisciplinarity, universities will need

to have the necessary flexibility to ensure postgraduate students receive the best supervisors, regardless of what department they happen to be located in.

Postdoctoral training: Mechanisms such as 'discipline hopper' fellowships (for example, as operated in the UK by the Medical Research Council) may be required.

3 Engage the public in science

The importance of science in addressing major global challenges of concern to society brings with it a concomitant need for much more effective engagement of the public in scientific issues and in informing science policy. The value and ultimate benefits of science to society are often unclear or questionable, and science generally remains shrouded in mystery to the majority of the public. The science community needs to find new ways to raise the importance of science both in the public arena and in terms of its importance for the political agenda.

Related to this is the broader matter of the ethical issues raised by developments in science. Some of the issues raised in the report will almost certainly give rise to new ethical concerns. Obvious examples include data on the genomes of individuals and their vulnerability to particular diseases, and the prediction of pandemics or natural disasters where little can be done to save the thousands of lives under threat. Computer and natural scientists will therefore need to call upon those with expertise in the area of public engagement with science in order to minimise the risk of the public becoming concerned about scientific 'scare stories' relating to the growing ability of scientists to manipulate nature. Addressing these matters in a prompt and transparent manner will help to increase the trust of the public in science and its importance, and their willingness, through their elected representatives, to ensure it is funded properly.

4 Re-think science funding and science policy structures

If policy makers are convinced by the arguments in this report, it would be relatively straightforward to make a case for significantly increased funding for science, whether in the UK, in Europe, the United States or elsewhere. However, we are not seeking to use this report to make such a case. Indeed, one might argue that questioning whether current funding structures work effectively is a necessary precursor to simply asking for more money to be spent within current structures. Indeed, perhaps more needs to be done to make the connection between funding and scientific output, and how we even measure scientific output. 'Fundamentally re-thinking' science funding and policy is typically classed in the 'can-of-worms' category that governments are wary of altering too much, but most in the science community in the UK, Europe and USA realise is something that needs to be done. We cannot cover all the key aspects of science funding and science policy here, and to recommend that yet some other kind of committee is established to look at this would be absurd when so many already exist. But something has to be done urgently to address the developments we outline here in order to help ensure that 2020 science is as exciting and fundamental to society as it needs to be. Some of the other recommendations here fall into that category, and are part of the science policy/science funding issue, and should be considered as such. However, some additional specific recommendations are:

Greater focus on interdisciplinary research, and new kinds of interdisciplinary collaborations to facilitate such research.

A greater focus on funding longer timescale projects so people work on solving problems, not flavour-of-the-month research areas that change the following year, which scientists then have to chase for funding.

Mechanisms to ensure these new kinds of science don't fall between cracks of traditional funding agencies and fail to get funding. This may mean a shake-up of traditional discipline - or branch-based (i.e. physical, biological, environmental) councils or agencies.

More risk-taking in science funding. We tend to expect that projects funded must always have a very high chance of 'success' but in doing so, funding agencies lean towards 'safe' research areas.

5 Create new kinds of research institutes

There is, of course, no shortage of research institutes focusing on everything from molecular biology to superconductivity, and from nanoscience to neuroscience. Many, however, tend to be highly discipline based (e.g. molecular biology). While such institutes are highly valuable, we argue there is a need for a new kind of institute focused on 'grand challenges' rather than 'grand disciplines'. Such new kinds of institutes would be highly interdisciplinary, combine teaching, training and research, and be focused on solving a problem or 'winning' a race, rather than simply producing papers. Two good examples of this kind of institute would be the Sanger Centre in the UK and the Earth Simulator in Japan.

6 Re-energise computer science to tackle 'grand challenges'

Computer science teaching and research is currently at an awkward crossroads where it needs to decide whether it is something that serves other disciplines, is an engineering exercise, or a real science in its own right. This report makes it clear that it can be a science in its own right. Its concepts and theorems are starting to prove fundamental in explaining natural and physical phenomena. However, clearly, there are significant aspects of computer science that are purely engineering. Both play a role in science. There is probably a good case to be made for calling each side separate disciplines. What is clear is that computer science needs to be re-energised in universities to inject new life

into the discipline, and to focus around helping find solutions to 'grand challenges', some of which we outline here, rather than having a tendency to focus on issues that have little connection to the real world, seen depressingly too often in computer science teaching and research.

7 A call to action to develop new conceptual and technological tools

Science policy makers should establish new dedicated programmes spanning science and technology to research and create the new kinds of conceptual and technological tools we outline in this report, and others we have not even begun to imagine. We believe this is absolutely vital. This will require a highly interdisciplinary focus, and the fostering of new kinds of communities (see the section 'New Kinds of Communities' in Part 2). The UK e-Science programme was a good example of what can be done in such programmes.

8 Develop innovative public private partnerships to accelerate science-based innovation

Governments, universities and businesses need to find new kinds of ways to work together. This is not new of course. In the UK, for example, the Government-commissioned Lambert Review made just such a recommendation after an extensive consultation with business and with universities, and the EU and others have been trying to foster closer academia–industry collaboration and partnerships. However, despite Governments all over Europe, as well as the USA and elsewhere looking to industry to increase their funding of public R&D, few examples of real industry–university collaborations reveal a rosy picture of mutual benefit. Too often, one or the other ends up being dissatisfied. We believe that entirely new kinds of public-private partnerships (PPPs) are needed in order to really accelerate science and science-based innovation. Such new kinds of PPPs are likely to take several forms, and all parties will need to experiment in this area. On industry's side, it needs to devise new models of R&D to remain competitive and in which universities are not a cheap and temporary source of 'contract work' but an essential, strategic partner in their ability to innovate and compete, in what has been termed an 'Open Innovation' model of R&D [61]. On academia's side, it needs to really raise its sights above just 'getting industry money', and also look beyond just producing papers (although this is vital for the healthy advancement of knowledge). On Government's part, science funding agencies need to be able to respond to changes in science and society and the economy quickly enough and with sufficient flexibility to enable industry to engage with universities in such new models of strategic R&D, rather than simply contractually. Establishing new kinds of joint research institutes between government, industry and the science community (see recommendation 5 above) is an interesting, and potentially highly mutually beneficial way forward.

9 Find better mechanisms to create value from intellectual property

Creating value from technology-based intellectual property (IP), whether created in industry or academia, has a reasonable, if sometimes chequered track record. In science-based innovation, creating value from intellectual property has proven more difficult, with the possible exception of the pharmaceutical sector. Perhaps this has been due in part to a focus on technology rather than science by the venture capital (VC) community. We believe that there is a need for both universities and industry (and probably governments) to find new and better ways to generate value from science-based IP. New approaches and concepts such as an 'eBay for IP' should be given consideration.

10 Use our findings

The beginning of this report makes clear two things. First, that this is just a first attempt to bring together some of the complex issues at the intersection of computing and science towards 2020. It is not a definitive statement but a pointer. Second, that one of the purposes of the report is to help inform and generate discussion about the future of science in the science community. If it helps to generate debate, dissent, ideas, better thought out arguments or indeed direction, it will have served this purpose.

The 2020 Science Group

References

[1] Wolfram S. A new kind of science. Champaign, IL: Wolfram Media Inc.; 2002.

[2] Stewart I. Nature's numbers: the unreal reality of mathematics. New York, NY: Basic Books; 1997.

[3] Moore GE. Cramming more components onto integrated circuits. Electronics 1965; 38(8):114–117.

[4] Ricadela A. Seismic Shift. InformationWeek http://www.informationweek.com/story/showArticle.jhtml?articleID=159400917. 2005. [Accessed 9 December 2005]

[5] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: a scalable peer-to-peer lookup service for internet applications. Proceedings of the SIGCOMM 2001 Conference, pp. 149–161. New York: ACM Press; 2001.

[6] Ferris C, Farrell J. What are Web services? Commun ACM 2003; 46(6):31.

[7] Andrews T, Curbera F, Dholakia H, Goland Y, Klein J, Leymann F *et al*. Business Process Execution Language for Web Services (BPEL4WS), Version 1.1. OASIS. http://ifr.sap.com/bpel4ws. 2003. [Accessed 6 December 2005]

[8] Emmerich W, Butchart B, Chen L, Wassermann B, Price SL. Grid service orchestration using the Business Process Execution Language (BPEL). J Grid Comput 2006 (in press). Available online at http://dx.doi.org/10.1007/s10723-005-9015-3. [Accessed 26 January 2006]

[9] Lesk M. http://archiv.twoday.net/stories/337419/ 2004. [Accessed 4 December 2005]

[10] Hey AJG, Trefethen AE. The data deluge: an e-Science perspective. In: Berman F, Fox GC, Hey AJG, editors. Grid computing: making the global infrastructure a reality. Chichester, UK: John Wiley & Sons Ltd; 2003, pp. 809–824.

[11] Bunn J, Newman H. Data intensive grids for high energy physics. In: Berman F, Fox G, Hey A, editors. Grid computing: making the global infrastructure a reality. Chichester, UK: John Wiley & Sons Ltd; 2003, pp. 859–906.

[12] Bell G, Gray J, Szalay A. Petascale computational systems: balanced cyberinfrastructure in a data-centric world. Letter to NSF Cyberinfrastructure Directorate. http://research.microsoft.com/~gray/papers/Petascale%20computational%20systems.pdf. 2005. [Accessed 6 December 2005]

[13] Gray J, Liu DT, Nieto-Santisteban M, Szalay AS, De Witt D, Heber G. Scientific data management in the coming decade. Technical Report MSR-TR-2005-10. Microsoft Research; 2005.

[14] Wing J. Computational thinking. Carnegie Mellon University, http://www.cs.cmu.edu/afs/cs/usr/wing/www/ct-paper.pdf. 2005. [Accessed 8 December 2005]

[15] Regev A, Shapiro E. Cellular abstractions: Cells as computation. Nature 2002; 419:343.

[16] Shapiro E. Algorithmic program debugging. Cambridge, MA: MIT Press; 1983.

[17] Popper KR. The logic of scientific discovery. London, UK: Hutchinson; 1959.

[18] Shapiro E. Inductive inference of theories from facts. Research Report 192. New Haven, CT: Department of Computer Science, Yale University; 1981.

[19] Shapiro E. In: Lassez J-L, Plotkin G, editors. Computational logic: essays in honour of Alan Robinson. Cambridge, MA: MIT Press; 1993, pp. 199–254.

[20] Cardelli L. Abstract machines of systems biology. In Transactions on Computational Systems Biology III . Lecture Notes in Computer Science Vol. 3737, 2005, pp. 145–168.

[21] Chaitin GJ. Information, randomness & incompleteness. Papers on algorithmic information theory. Series in Computer Science – Vol. 8. Singapore: World Scientific Publishing Co. Pte Ltd; 1987.

[22] Huang CY, Ferrell JE. Ultrasensitivity of the mitogen-activated protein kinase cascade. Proc Natl Acad Sci USA 1996; 93(19):10078–10083.

[23] Phillips A, Cardelli L. A graphical representation for the stochastic pi-calculus. BioConcur'05. http://research.microsoft.com/Users/luca/Papers/A%20Graphical%20Representation%20for%20Stochastic%20Pi-calculus%20(BioConcur).pdf. 2005. [Accessed 20 January 2006]

[24] Cootes AP, Muggleton SH, Sternberg MJ. The automatic discovery of structural principles describing protein fold space. J Mol Biol 2003; 330(4):839-850.

[25] Sternberg MJE, Muggleton SH. Structure activity relationships (SAR) and pharmacophore discovery using inductive logic programming (ILP). QSAR Comb Sci 2003; 22:527–532.

[26] Muggleton SH, Lodhi H, Amini A, Sternberg MJE. Support vector inductive logic programming. In: Proceedings of the 8th International Conference on Discovery Science, Lecture Notes in Artificial Intelligence Vol. 3735, 2005, pp. 163–175. Springer-Verlag.

[27] Zytkow JM, Zhu J, Hussam A. Automated discovery in a chemistry laboratory. In: Proceedings of the 8th National Conference on Artificial Intelligence. Boston, MA. AAAI Press/MIT Press; 1990, pp. 889–894.

[28] King RD, Whelan KE, Jones FM, Reiser PKG, Bryant CH, Muggleton SH, Kell DB, Oliver SG. Functional genomic hypothesis generation and experimentation by a robot scientist. Nature 2004; 427:247–252.

[29] Karlsson M, Davidson M, Karlsson R, Karlsson A, Bergenholtz J, Konkoli Z *et al*. Biomimetic nanoscale reactors and networks. Annu Rev Phys Chem 2004; 55:613–649.

[30] Kobayashi H, Kaern M, Araki M, Chung K, Gardner TS, Cantor CR, Collins JJ. Programmable cells: Interfacing natural and engineered gene networks. Proc Natl Acad Sci USA 2004; 101(22):8414–8419.

[31] Basu S, Mehreja R, Thiberge S, Chen M-T, Weiss R. Spatiotemporal control of gene expression with pulse-generating networks. Proc Natl Acad Sci USA 2004; 101(17):6355–6360.

[32] Nivens DE, McKnight TE, Moser SA, Osbourn SJ, Simpson ML, Sayler GS. Bioluminescent bioreporter integrated circuits: Potentially small, rugged and inexpensive whole-cell biosensors for remote environmental monitoring. J Appl Microbiol 2004; 96:33–46.

[33] von Neumann J, Burks AW. Theory of self-reproducing automata, Urbana, IL: University of Illinois Press; 1966.

[34] Drexler KE. Molecular engineering: An approach to the development of general capabilities for molecular manipulation. Proc Natl Acad Sci USA 1981; 78(9):5275–5278.

[35] Penrose LS, Penrose R. A self-reproducing analogue. Nature 1957; 4571:1183.

[36] Feynman RP. There's plenty of room at the bottom: an invitation to open up a new field of physics. Eng Sci 1960; 23(5):22–36.

[37] Benenson Y, Gil B, Ben-Dor U, Adar R, Shapiro E. An autonomous molecular computer for logical control of gene expression. Nature 2004; 429:423–429.

[38] Endy D. Foundations for engineering biology. Nature 2005; 438(24):449–453.

[39] Luisi PL, Ferri F, Stano P. Approaches to semi-synthetic minimal cells: a review. Naturwissenschaften 2005; 15:1–13.

[40] Szyperski C, Gruntz D, Murer S. Component software. 2nd ed. New York: Addison-Wesley; 2002.

[41] Messerschmitt D, Szyperski C. Software ecosystem: understanding an indispensable technology and industry. Cambridge, MA: MIT Press; 2003.

[42] Acheson A, Bendixen M, Blakeley JA, Carlin P, Ersan E, Fang J *et al*. Hosting the .NET Runtime in Microsoft SQL Server. Proceedings of the ACM SIGMOD Conference 2004, pp. 860–865. New York: ACM Press; 2004.

[43] Gonthier, G. A computer-checked proof of the four colour theorem. Microsoft Research Cambridge, http://research.microsoft.com/~gonthier/4colproof.pdf. 2005. [Accessed 7 December 2005]

[44] Burnett M, Cook C, Rothermel G. End-user software engineering. Commun ACM 2004; 47(9):53–58.

[45] Szyperski C. The making of a software engineer: challenges for the educator. International Conference on Software Engineering (ICSE'05). Proceedings of the 27th International Conference on Software Engineering; 2005, pp. 635–636.

[46] Loh J, Wackernagel M. Living planet report 2004. Gland, Switzerland: World Wide Fund for Nature; 2004.

[47] Millennium Ecosystem Assessment. Ecosystems and human well-being: synthesis report. Washington, DC: Island Press; 2005.

[48] Levin SA, Grenfell B, Hastings A, Perelson AS. Mathematical and computational challenges in population biology and ecosystems science. Science 1997; 275:334.

[49] Soberon J, Peterson AT. Biodiversity informatics: managing and applying primary biodiversity data. Phil Trans R Soc (B) 2004; 35:689–698.

[50] Peterson AT, Ortega-Huerta MA, Bartley J, Sanchez-Cordero V, Soberon J, Buddemeier RH, Stockwell DR. Future projections for Mexican faunas under global climate change scenarios. Nature 2002; 416:626–629.

[51] Lovejoy TE, Hannah L, editors. Climate change and biodiversity. New Haven, CT: Yale University Press; 2005.

[52] Gaston JK, O'Neill M. Automated species identification: why not? Phil Trans R Soc (B) 2004; 359:655–667.

[53] Grimm V, Wzyomirsky T, Aikman D, Uchmanski J. Individual-based modelling and ecological theory: synthesis of a workshop. Ecol Modell 1999; 115(2):275–282.

[54] Lund O, Nielsen M, Lundegaard C, Kesmir C, Brunak S. Immunological bioinformatics. Cambridge, MA: MIT Press; 2005.

[55] Vangrevelinghe E, Zimmermann K, Schoepfer J, Portmann R, Fabbro D, Furet P. Discovery of a potent and selective protein kinase CK2 inhibitor by high-throughput docking. J Med Chem 2003; 46(13):2656–2662.

[56] Spergel DN, Verde L, Peiris HV, Komatsu E, Nolta MR, Bennett CL *et al*. First-Year Wilkinson Microwave Anisotropy Probe (WMAP) observations: determination of cosmological parameters. Astrophys J Suppl 2003; 148:175–194.

[57] Springel V, White SDM, Jenkins A, Frenk CS, Yoshida N, Gao L *et al*. Simulations of the formation, evolution and clustering of galaxies and quasars. Nature 2005; 435:629–636.

[58] Crick FH, Orgel LE. Directed Panspermia. Icarus 1973; 19:341–346.

[59] Zauner K-P. Molecular information technology. Crit Rev Solid State Mater Sci 2005; 30(1):33–69.

[60] Lehn J-M. Supramolecular chemistry: from molecular information towards self-organization and complex matter. Rep Prog Phys 2004; 67:249–265.

[61] Chesbrough HW. Open innovation: the new imperative for creating and profiting from technology. Boston: Harvard Business School Press; 2003.

Glossary

Actionable patterns: Knowledge that can be uncovered in large complex databases and can act as the impetus for some action. These actionable patterns are distinct from the lower value patterns that can be found in great quantities and with relative ease through so-called data dredging.

AIDS: See HIV.

Algebra: A branch of mathematics which may be characterised as a generalisation and extension of arithmetic, in which symbols are employed to denote operations, and to represent variables such as number and quantity. In a more abstract sense, a calculus of symbols combining according to certain defined laws.

Bandwidth: A measure of the capacity of a communications channel. The higher a channel's bandwidth, the more information it can carry. Bandwidth is usually stated in bits per second (bps), kilobits per second (kbps), megabits per second (Mbps), and increasingly in gigabits per second (Gbps). Confusingly, sometimes measured in bytes per second, e.g. gigabytes per second (GBps).

Bayesian statistics: Statistics which incorporates prior knowledge and accumulated experience into probability calculations. The name comes from the frequent use of Bayes' theorem in this discipline. Bayesian statistics are used in Bayesian networks, graphical models that depict the probability of events or conditions based on causal relations.

Beowulf cluster: A high-performance parallel computing cluster on inexpensive personal computer hardware – usually identical PC computers running an open source Unix®-like operating system.

Big Bang: See hot Big Bang.

Bioinformatics: A branch of science concerned with the use of techniques from applied mathematics, informatics, statistics, and computer science to solve biological problems. Research in computational biology often overlaps with systems biology. Major research efforts in the field include sequence alignment, gene finding, genome assembly, protein structure alignment, protein structure prediction, prediction of gene expression and protein–protein interactions, and the modelling of evolution. Often referred to as computational biology.

BPML (Business Process Modelling Language): See workflow.

BSE (bovine spongiform encephalopathy): Commonly known as mad cow disease, BSE is a fatal, neurodegenerative disease of cattle. The causative agent of BSE is thought to be a specific type of protein: a misshaped prion protein which carries the disease between individuals and causes deterioration of the brain.

Calculus (pl. calculi): In mathematics, calculus most commonly refers to elementary mathematical analysis, which investigates motion and rates of change. Calculi also describe a system for defining functions, proving equations between expressions, and calculating values of expressions (e.g. λ-calculus, π-calculus and other process calculi).

CAPTCHA: An acronym for Completely Automated Public Turing test to Tell Computers and Humans Apart. In essence, a program that can generate and grade tests that most humans can pass but current computer programs fail, for example recognising words displayed as distorted text.

CBM (Cosmic Microwave Background): A form of electromagnetic radiation that radiates throughout the Universe in the microwave range. CMB is the dominant component of cosmic background radiation (CBR) that includes other cosmological backgrounds (e.g. infrared, radio, X-ray, gravity-wave, neutrino). The CMB is the cooled remnant of the hot Big Bang.

CERN: The Centre Européen pour la Recherche Nucléaire (European Organization for Nuclear Research) is the world's leading laboratory for particle physics. It has its headquarters in Geneva. Its flagship project is the Large Hadron Collider, a particle accelerator due to switch on in 2007.

Chromosome: Highly-packaged DNA which carries the genetic information in biological cells. Normal human cells contain 23 pairs of chromosomes including the sex-chromosome pair (XY in males and XX in females).

CJD (Creutzfeldt-Jakob Disease): A rare and incurable brain disease. Like BSE, CJD is caused by a unique infectious agent called a prion. CJD usually affects people aged 45–75, most commonly appearing in people between the ages of 60 and 65. The more recently recognised vCJD (variant CJD) occurs in younger people.

Client-server: a network application architecture which separates the client (usually the graphical user interface) from the server. Each instance of the client software can send requests to a server. Server software generally, but not always, runs on powerful computers dedicated to running the application – application servers (cf. peer-to-peer).

Commodity machines: 'Off-the-shelf' computer hardware as opposed to custom-built computers. Such machines are mass produced and therefore cheap and standardised which makes them suitable for use as units in computer clusters (see Beowulf cluster).

Complex systems: Formally a system of many parts (components) which are coupled in a nonlinear fashion. Because such systems often exhibit complex behaviour belied by the apparent simplicity of their components, they are said to possess 'emergent' properties. Examples of complex systems include the immune system, the stock-exchange and weather systems.

Composability: A system design principle that deals with the inter-relationships of components. A highly composable system provides recombinant components that can be selected and assembled in various combinations to satisfy specific user requirements. Composable components are both self-contained (i.e. can be deployed independently) and stateless (i.e. treats each request as an independent transaction).

Concurrency: The sharing of common resources between computations which execute overlapped in time (including running in parallel).

Deterministic process: A process that behaves predictably, i.e. given a particular input, a deterministic process will always produce the same correct output, and the underlying machine will always pass through the same sequence of states.

DNA (deoxyribonucleic acid): A macromolecule which carries the genetic material of nearly all forms of life. DNA is a polymer of nucleotides (compounds composed of one nitrogenous base, one phosphate molecule, and one sugar molecule). See also genome; RNA.

e-Science: Large-scale science carried out through distributed global collaborations enabled by the Internet.

Exabytes: See storage capacity.

FLOPS (Floating-Point Operations Per Second): a common benchmark measurement for rating the speed of microprocessors, hence megaFLOPS (MFLOPS, 10^6 FLOPS), gigaFLOPS (GFLOPS, 10^9 FLOPS), teraFLOPS (TFLOPS, 10^{12} FLOPS), and petaFLOPS (PFLOPS, 10^{15} FLOPS).

FMD (foot-and-mouth disease): A highly contagious, but rarely fatal, viral disease of cattle and pigs.

Gene: The fundamental physical and functional unit of heredity. A gene is an ordered sequence of nucleotides (one of four possible molecules identified as A, T, C or G) located in a particular position on a particular chromosome that encodes a specific functional product (usually a protein).

Genome: The DNA complement of an organism, usually one or more chromosomes, and in the case of some organisms, additional non-chromosomal DNA (such as plasmids, short circular DNA structures in bacteria). The one exception is RNA viruses in which the genome is composed of RNA.

Genomics: See 'omics.

Grid computing: The use of the resources of many separate computers connected by a network (usually the Internet) to solve large-scale computation problems. Grid computing involves the distribution of computer power and data storage capacity. Although some deployments within single organisations have been labelled as grids, a key goal of grid computing is to provide transparent access to resources across administrative domains.

Higgs boson: A hypothetical elementary particle predicted to exist by the Standard Model of particle physics. According to the Standard Model, the Higgs boson is a component of the Higgs field which is thought to permeate the Universe and to give mass to other particles, and to itself. As of 2005, no experiment has definitively detected the existence of the Higgs boson, but it is expected that the Large Hadron Collider at CERN will be able to confirm or disprove its existence.

HIV (Human Immunodeficiency Virus): A virus that primarily infects vital components of the human immune system. The progressive destruction of the immune system leads to the onset of the Acquired Immunodeficiency Syndrome (AIDS).

Hot Big Bang: A model of the origin of the Universe, in which space, time and matter emerged from a primordial state at very high density and temperature, potentially a singularity of the type found inside black holes. Data from a variety of sources, including the Cosmic Microwave Background, the abundance of elements in the Universe, the large scale structure of galaxies, and particle physics experiments are all consistent with the Λ-CDM version of the hot big bang. A more complete understanding of the initial state awaits the development of a theory of quantum gravity (a goal of superstring theory), and experimental measurements of the fields (such as the Higgs field) which populate the quantum vacuum.

HPC (High Performance Computing): Computationally demanding applications deployed on powerful hardware with multiple processors, shared memory and very low latency. Usually, computer systems in or above the teraFLOPS region are counted as HPC computers.

Immune system: A system of specialised cells and organs evolved to protect an organism from external biological influences, in particular, disease-causing pathogens.

Inductive Logic Programming (ILP): A machine learning strategy where, from a database of facts and expected results divided into positive and negative examples, a system tries to derive a logic program that proves all the positive and none of the negative examples.

Interoperability: The capability of different programs to read and write the same file formats and use the same protocols.

Kolmogorov complexity: A measure of complexity chiefly describing the complexity of a string as being measured by the length of the shortest program required to generate it.

Lab-on-chip technology: Miniaturised (silicon) chips technology combining microfluidics, microelectronics and biochemical and molecular biology to perform laboratory analyses (often for biomedical applications such as disease diagnosis).

Λ-CDM (Lambda-Cold Dark Matter) model: The current concordance model of the Big Bang and the simplest model in agreement with cosmic microwave background, large-scale structure and supernovae observations. It includes a cosmological constant (Λ) which causes the expansion of the Universe to accelerate, as well as an unknown form of matter, which is not observed by optical, infrared or radio telescopes, whose existence is inferred from its gravitational effects. The model does not require significant amounts of hot dark matter, from high energy particles.

LAN (Local Area Network): A communication network for computers covering a local area, like a home, office or small group of buildings such as a college (cf. WAN).

Latency: The delay before a transfer of data begins following an instruction for its transfer.

LHC (Large Hadron Collider): A particle accelerator and collider located at CERN and scheduled to start operation in 2007. It will become the world's largest particle accelerator. Physicists hope to use the collider to clarify many fundamental physics issues such as the proposed existence of the Higgs boson, the nature of the quantum vacuum and the number of space dimensions.

Machine learning: A branch of computer science concerned with developing methods for software to learn from experience or extract knowledge from examples in a database. Applications of machine learning include detecting credit card fraud, classifying DNA sequences and speech and handwriting recognition.

Macromolecule: A molecule with a large molecular mass. Generally, the use of the term is restricted to polymers and molecules which structurally include polymers. Many examples come from biology including proteins (amino acid polymers), starches (sugar polymers) and nucleic acids (such as DNA, a nucleotide polymer).

Managed runtime: An execution environment that provides a number of services for the controlled execution of programs and components. Services could include, but are not limited to: (i) type system, (ii) metadata faculties (i.e. reflection), (iii) security system (i.e. verification of code), (iv) memory management, (v) system libraries. The Java™ Virtual Machine and Common Language Runtime are examples of managed runtimes.

Message passing: A programming technique (used for example in concurrent programming, parallel programming, and object-oriented programming) where instead of using shared memory and locks or other facilities of mutual exclusion, different threads of execution communicate via passing messages. Prominent models of computation based on message passing include the process calculi.

Microfluidics: A multidisciplinary field comprising physics, chemistry, engineering and biotechnology that studies the behaviour of fluids at volumes thousands of times smaller than a common droplet and by extension concerns the design of systems in which such small volumes of fluids will be used (see lab-on-chip technology).

Modelling: The process of generating a model as a conceptual representation of some phenomenon (e.g. global warming models).

Molecular computer: An autonomous computing entity (i.e. one controllably capable of processing an input into a specific output) constructed using macromolecules such as DNA or protein.

Moore's law: A rule of thumb in the computer industry about the growth of computing power over time named after Gordon Moore following the publication of his 1965 article 'Cramming more components onto integrated circuits'. Moore's law states that the growth of computing power (specifically as measured by the number of transistors per unit of area on a silicon chip) follows an exponential law. The 'doubling' period in Moore's paper was originally 12 months, and is now variously and inconsistently defined. Nevertheless, the spirit of the law is generally considered to have held true to date. It is now thought that the limits of Moore's law (at least in the *sensu strico* terms of transistor miniaturisation) will soon be reached.

MPI (Message Passing Interface): A computer communications protocol and the *de facto* standard for communication among the nodes running a parallel program on a distributed memory system (see message passing).

Multicore processor: A chip with more than one processing units (cores). Mostly, each unit can run multiple instructions at the same time and each has its own cache. Such an architecture is seen as the key to maintain the current rate of computational power improvement.

Nanotechnology: The branch of technology that deals with dimensions and tolerances of 0.1–100 nanometres, or, generally, with the manipulation of individual atoms and molecules and by extension, the science of miniaturised machines within such size ranges.

Network theory: A branch of applied mathematics concerned with the idea of a graph as a representation of a network. As such, network theory deals with the same general subject matter as graph theory but is led by applications, often to computer networks but certainly not limited to those (for example, network theory also applies to gene regulatory network and social networks).

Neural network: an interconnected group of artificial (software or hardware) neurons and by extension, a machine learning technique based on the observed behaviour of biological neurons. Neural networks consist of a set of elements that start out connected in a random pattern, and, based upon operational feedback, are moulded into the pattern required to generate the required results. Neural networks are used in applications such as robotics, diagnosing, forecasting, image processing and pattern recognition.

Neuron: The primary cell of the nervous system, also known as nerve cells. Neurons communicate through synapses, specialised junctions through which cells of the nervous system signal to one another and to non-neuronal cells such as muscles or glands. The human brain has about 100 thousand million (10^{11}) neurons and 100 trillion (10^{14}) synapses between them.

Non-deterministic: A property of a computation which may have more than one result (cf. deterministic process and stochastic process).

OLAP (On-Line Analytical Processing): A category of applications and technologies for collecting, managing, processing and presenting multi-dimensional data for analysis and management purposes.

-omics: -omics is a suffix commonly attached to biological subfields for describing very large-scale data collection and analysis; by extension, 'omics is an umbrella terms for all such datasets. -omics refers to the whole 'body' of some definable entities, hence genomics [the genetic (DNA) complement of a cell or organism], transcriptomics [the set of all expressed genes (i.e. genes transcribed into RNA) in a cell or organism] and proteomics [all the proteins from a cell or organism].

Organismic biology: The study of biological entities above the cell level. Organismic biology spans from the ecology of single populations to that of the whole biosphere, and from micro-evolutionary phenomena to palaeontology, phylogenetics and macroevolution.

Pathogen: A disease-causing biological agent. Pathogens can be bacteria [e.g. *Mycobacterium tuberculosis* (tuberculosis)], viruses [e.g. HIV (AIDS)], protozoa [e.g. *Plasmodium falciparum* (malaria)], fungi [e.g. *Tinea pedis* (athlete's foot)], multi-cellular parasites [e.g. tapeworm] and prions [e.g. BSE].

Peer-to-peer (or P2P): A network computing model in which all computers are treated as equals on the network. A pure peer-to-peer network does not have the notion of clients or servers, but only equal peer nodes that simultaneously function as both clients and servers to the other nodes on the network (cf. client-server).

Power-law: A relationship between two quantities such that the magnitude of one is proportional to a fixed power of the magnitude of the other (mathematically: $y = ax^k$). In a scale-free network, the probability of an existing node connecting to a new node attached to the network is dictated by a power-law (see scale-free network).

Primitive types (or primitives): Datatypes provided by a programming language as basic building blocks. Depending on the language and its implementation, primitive types may or may not have a one-to-one correspondence with objects in the computer's memory (e.g. integer, character, boolean).

Process calculus (pl. process calculi): Process calculi (or process algebras) allow high-level descriptions of interaction, communication and synchronisation in concurrent systems. They include features such as events, prefix, choice, communication via channels, etc. Algebraic laws allow formal reasoning about equivalent processes. Examples include the π-calculus for mobile processes. Such calculi have also been successfully applied to model a number of complex systems such as biological networks.

Programming platforms: Some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating system, or programming languages and their runtime libraries. The .NET™ framework and the Java™ platform are examples of software development platforms.

Protein: A large macromolecule made up of one or more chains of amino acids (small nitrogen containing organic molecules). Proteins are the principal constituents of living cells and serve as enzymes, hormones, structural elements and antibodies.

Proteomics: See -omics.

Relational algebra: An algebra used to define part of the data manipulation aspect, the relational model – essentially a model where data in tabular format can be joined according to certain constraints and used in many database management systems such as Oracle® and Microsoft® SQL Server. Relational algebra operations are often used in database query optimisation as an intermediate representation of a query to which certain rewrite rules can be applied to obtain a more efficient version of the query.

RNA (ribonucleic acid): A macromolecule which, like DNA, is a polymer of nucleotides (but of a slightly different chemical nature than the nucleotides of DNA). These molecules are involved in the transfer of information from DNA and in the process of protein synthesis. See also DNA; genome.

RPC (Remote Procedure Call): A protocol that allows a computer program running on one host to cause code to be executed on another host without the programmer needing to explicitly code for this.

SARS (Severe Acute Respiratory Syndrome): An atypical form of pneumonia caused by a virus.

Scale-free network: A network in which a few nodes (know as hubs) are highly connected, but most other nodes are not very connected. Such networks are very resilient to random node failures, but deliberate attacks on hubs can easily debilitate them. A multitude of real-world networks have been shown to be scale-free, including many kinds of computer networks (e.g. the World Wide Web) and biological networks (e.g. metabolic networks and protein–protein interaction networks).

Secure runtime: See managed runtime.

Semantic web: A project that intends to create a universal medium for information exchange by giving meaning (semantics), in a manner understandable by machines, to the content of documents on the Web. Such an environment would permit the unification of all scientific content by computer languages and technologies to exploit the interrelationships between scientific concepts.

Smart drugs: Products of the drug discovery process characterised by one or more of the following properties: (i) the use of smarter ways to develop drugs that are safe, effective, and affordable (such as computational drug–target interaction modelling); (ii) drugs that are themselves smart in some sense, e.g. targeted, adaptive, and so forth; and (iii) drugs that improve human cognitive abilities (nootropic drugs), either by heading off age- or disease-related declines, restoring lost function, or potentially even enhancing native abilities associated with intelligence.

SOA (service-oriented architecture): A software architectural concept that defines the use of services to support the requirements of software users. In a SOA environment, resources are made available to other participants in the network as independent services that the participants access in a standardised way. Web services are an implementation of SOA-based technology.

Software artefacts: Products of software development, for example specifications, designs, or code.

Standard Model: In particle physics, a theory which describes the strong, weak, and electromagnetic fundamental forces, as well as the fundamental particles that make up all matter.

Stochastic process: A process that follows some random probability distribution or pattern, so that its behaviour may be analysed statistically but not predicted precisely (e.g. stock market and exchange rate fluctuations, Brownian motion; cf. deterministic process).

Storage capacity: The total amount of stored information that a storage device or medium can hold. Common measures are usually given in multiples of bytes, however, there is inconsistency in the naming conventions. For example, a megabyte (MB) can be 1,000,000 bytes (1000^2, 10^6) or 1,048,576 bytes (1024^2, 2^{20}). Similarly, a terabyte (TB) is 10^{12} or 2^{40} bytes, a petabyte (PB) 10^{15} or 2^{50} bytes and an exabyte 10^{18} or 2^{60} bytes. A mebibyte (MiB) is however unambiguously 2^{20} bytes, and likewise tebibyte (TiB, 2^{40} bytes), pebibyte (PiB, 2^{50}) and exbibyte (EiB, 2^{60} bytes).

Supersymmetry: A particle physics theory according to which every particle has a supersymmetric partner – a particle with the same properties, but differing in the amount of angular momentum it carries. Although supersymmetry has yet to be observed in the real world, it remains a vital part of many proposed theories of physics, including various extensions to the Standard Model and superstring theory, the current best candidate for a unified 'Theory of Everything'.

Synapse: See neuron.

Synthetic biology: The design and construction of new biological parts, devices, and systems as well as the re-design of existing, natural biological systems for useful purposes (e.g. biological photographic film).

Systems biology: A multi-disciplinary field that seeks to integrate different levels of information to understand how biological systems function. Systems biology investigates the relationships and interactions between various parts of a biological system (e.g. metabolic pathways, cells, organs). Much of current systems biology is concerned with the modelling of the 'omics to generate understandable and testable models of whole biological systems, especially cells.

Terabyte: See storage capacity.

TeraFLOPS: See FLOPS.

Theranostics: Combining a therapeutic entity (e.g. drug course) with a corresponding diagnostic test ensuring that the right treatment is used for the right patient, at the right time. Such a strategy is particularly relevant to personalised medicine where treatment (including prophylactic treatment), is specifically tailored to the individual (for example, with respect to their genetic make-up).

Transcriptomics: See 'omics.

Turing machine: A notional computing machine for performing simple reading, writing, and shifting operations in accordance with a prescribed set of rules. Despite its simplicity, a Turing machine can be adapted to simulate the logic of any computer that could possibly be constructed, and studying its abstract properties yields many insights in computer science and complexity theory. A universal Turing machine is a Turing machine able to simulate any other Turing machine.

Universal Turing machine: See Turing machine.

Von Neumann architecture: A computer design model that uses a single storage structure to hold both instructions and data.

WAN (Wide Area Network): A communications network that is capable of spanning a large geographic area (generally larger than a metropolitan area) using phone lines, microwaves, satellites, or a combination of communication channels as well as specialised computers to connect several smaller networks (cf. LAN).

Web services: See SOA.

Workflow: the sequence of processes representing a task from initiation to completion. Scientific workflows are mostly concerned with throughput of data through various algorithms, applications and services, whilst business workflows tend to concentrate on scheduling task executions, ensuring dependences which are not necessarily data-driven and may include human agents. Workflows can be scripted in languages such as the Business Process Modelling Language (BPML), and often involve interaction with, and integration between, services such as web services.

This report is printed on a mix of recycled and sustainable forestry paper,
and produced using clean production methods and chlorine-free processes.