Advance Access publication May 28, 2006

Agents in bioinformatics, computational and systems biology

Emanuela Merelli, Giuliano Armano, Nicola Cannata, Flavio Corradini, Mark d'Inverno, Andreas Doms, Phillip Lord, Andrew Martin, Luciano Milanesi, Steffen Möller, Michael Schroeder and Michael Luck

Abstract

The adoption of agent technologies and multi-agent systems constitutes an emerging area in bioinformatics. In this article, we report on the activity of the Working Group on Agents in Bioinformatics (BIOAGENTS) founded during the first AgentLink III Technical Forum meeting on the 2nd of July, 2004, in Rome. The meeting provided an opportunity for seeding collaborations between the agent and bioinformatics communities to develop a different (agent-based) approach of computational frameworks both for data analysis and management in bioinformatics and for systems modelling and simulation in computational and systems biology. The collaborations gave rise to applications and integrated tools that we summarize and discuss in context of the state of the art in this area. We investigate on future challenges and argue that the field should still be explored from many perspectives ranging from bio-conceptual languages for agent-based simulation, to the definition of bio-ontology-based declarative languages to be used by information agents, and to the adoption of agents for computational grids.

Keywords: *agents; multi-agent systems; data analysis and management; biological systems modelling and simulation; grid computing; semantic web; web services; LIMS; bioinformatics; computational biology; systems biology*

Corresponding author. Emanuela Merelli, Dipartimento di Matematica e Informatica Via Madonna delle Carceri 9 62032 Camerino, Italy, Tel: +39 0737 402567; Fax: +39 0737 402565; E-mail: emanuela.merelli@unicam.it

Emanuela Merelli is a researcher in Computer Science at the University of Camerino in Italy and member of the COmplex SYstem Research Group. Her main interests lie in applying formal, semiformal and agent-based methods for designing software systems with applications in bioinformatics and systems biology.

Giuliano Armano is currently an Associate Professor of computer engineering at the University of Cagliari, leading the IASC ('Intelligent Agents and Soft-Computing') group. His educational background ranges over machine learning and software agents. The above research topics are mainly experimented in the field of bioinformatics.

Nicola Cannata is a Postdoctoral researcher in the COmplex SYstem Research Group, Department of Mathematics and Computer Science, University of Camerino. His main research interests include bioinformatics, computational and systems biology.

Flavio Corradini is a Full Professor in Computer Science at the University of Camerino, Italy. He is the leader of the COmplex SYstem Research Group involved in studying formal methods for modelling complex systems for different application domains. He has published many books and papers on formal methods and concurrent systems.

Mark d'Inverno is a Professor in Computer Science at the University of Westminster. His research has been concerned with developing mathematical models of intelligent agent and multi-agent systems, which he is now applying in a number of different contexts including biology, music, art and design.

Andreas Doms is a PhD student at the Biotec of TU Dresden and a software architect at Transinsight. His interest are biomedical textmining and semantic web technologies.

Phillip Lord is a Lecturer at the School of Computing Science, University of Newcastle-Upon-Tyne. His research interests lie in applying new computing technologies to biology

Andrew Martin received his degree and DPhil from the University of Oxford. He is the Senior Lecturer in Bioinformatics at UCL with particular interests in antibody sequence and structure, protein modelling and the effects of mutations on protein structure and function.

Luciano Milanesi is a Researcher at the Italian National Research Council (CNR). He is the head of the Bioinformatics Division of CNR-ITB. He has been Principal Investigator of many European Project. He has published more than 140 contributions in books and scientific publications in bioinformatics.

Steffen Möller teaches Bioinformatics at the University of Lübeck, Germany. His research interests in agent technology have their roots in the information integration for automated sequence annotation and the collection of evidence for putatively disease-associated genes in statistical genetics.

Michael Schroeder is a Professor in Bioinformatics at the Biotec of TU Dresden and CSO of Transinsight. His main interests are structural protein interactions and ontology-based literature search.

Michael Luck is a Professor of Computer Science at the University of Southampton in the UK. He has published eight books and over 150 papers on aspects of agents and multi-agent systems, including in relation to the application of agents to bioinformatics.

INTRODUCTION

The avalanche of data that has been generated, particularly in biological sequences and more recently also in transcriptional and structural data, interactions and genetics, has led to the early adoption of tools for unsupervized automated analysis of biological data during the mid-1990s [1, 2]. Computational analysis of such data has become increasingly more important, and many more tools and models for the interpretation of biological data have been developed in recent years. However, not all of these are publicly available or permit bulk submissions via the web. Moreover, some tools require training, particularly for individual organisms, and tools may be mutually interdependent.

The reason for establishing a Working Group on Agents in Bioinformatics (BIOAGENTS) was to achieve improvements in the field of bioinformatics by designing and implementing new flexible information and communication technologies tools able to support biological science data analysis and to distribute, at least partially, the computation burden, while reducing the need for the transfer of huge amounts of data. From this perspective, it is believed that software agents can play a major role. The scope of the Working Group was to promote a collaboration between the agent and bioinformatics communities, with the aim of creating synergies for modelling complex biological systems. As suggested by the AgentLink II Roadmap [3], one of the most promising emerging application domains for agent technologies is the biological sciences, both in relation to multi-agent systems for simulating and modelling biological systems, and for supporting the automation of information-gathering and information-inference processes.

AgentLink III (http://www.agentlink.org) was a coordination action for agent-based computing, funded by the European Commission's 6th Framework Programme, which provides support for researchers and developers with a common interest in agent-based computing. One of the most important and visible activities of AgentLink III has been concerned with organizing a periodic Technical Forum (AL3-TF), in which Technical Forum Groups (TFGs) meet to discuss issues of key interest to the agent community. In 2004, AgentLink organized the first Technical Forum, held from the 30th of June to the 2nd of July, in Rome, Italy, at which the first BIOAGENTS Technical Forum Group founded the Working Group [4].

Here, we report on the state of the art in using agents in bioinformatics, presenting the activities and results of the Working Group and future perspectives. In particular, subsequent sections first introduces 'Agents and multi-agent systems', then motivates the use of 'Agents in bioinformatics' by discussing recent experiences within BIOAGENTS & points out 'Future challenges' of agents in bioinformatics & finally concludes the article.

AGENTS AND MULTI-AGENT SYSTEMS

Agents can be considered as a distinct kind of software abstraction, in the same way that methods, functions and objects are software abstractions. More specifically, an agent is a high-level software abstraction that provides a convenient and powerful way to describe a complex software entity in terms of its behaviour within a contextual computational environment. It differs from an object in the capability to control its own state. The *weak* notion of agents is of flexible problem-solving computational entities that are reactive (respond to the environment), proactive (maintain overarching goals), autonomous (not externally controlled) and interact with other such entities. By contrast, the *strong* notion views agents as composed of particular mental or cognitive abilities, suggesting agent architectures based on the belief–desire–intention model. The weak form has relatively low-level agents that do little computation processing, and the outcomes emerge from the results of the interactions of large numbers of agents. The coarser grained agents yield higher-level of communication and stronger individual problem-solving capabilities.

Building on this premise, *multi-agent systems*, in which multiple agents interact in some overarching system architecture, have been argued to be a particularly suitable level and means of abstraction for solving complex problems. This is achieved through the modelling and engineering of complex systems [5], which are characterized by organization structures and coordination processes that are increasingly better articulated and more dynamic than alternative forms [6].

In this view, an agent is a computer system capable of flexible, autonomous *problem-solving* actions; it is capable of operating as a stand-alone

process, and performing actions without user intervention by maintaining a description of its own processing state and the state of environment in which it is situated. The environments in which agents operate are typically dynamic, open, unpredictable and populated by other agents. An agent must therefore also be able to communicate with other agents and the execution environment itself [7, 8].

A *communication act* between two agents is facilitated if a suitable ontology exists, shared by both agents. The communication itself is distinct, however. For example, agent communication languages such as KQML [9] or FIPA ACL (http://www.fipa.org) provide the performatives (or types of message) that may be required here, but both allow for the specification of a particular ontology to ensure that the content of the message is understood by both the parties. The use of ontologies guarantees agreement on the semantics of the exchanged data. Moreover, whenever an agent acquires additional information, it can integrate it with its personal knowledge base. Each agent is responsible for the consistency and the correctness of this operation.

Agents provide designers and developers with a way of structuring an application around autonomous, communicative elements, and lead to the construction of software tools and infrastructure to support the design metaphor. In this sense, they offer a new and often more appropriate route to the development of complex systems. In order to support this view of systems development, particular tools and techniques need to be used: from agent computing platforms to support the design and engineering of agents and multi-agent systems, to more general infrastructures supporting the integration of current technologies, such as web services. However, agent *technologies* are distinct in spanning a range of specific techniques and algorithms for dealing more specifically with interactions with others in these dynamic and open environments. Such techniques include those for learning from, and about, other agents in the environment and user preferences, finding ways to negotiate and cooperate with agents, and developing appropriate means of forming and managing coalitions. With the increasing prevalence of agent-based computing in recent years, research on agent-oriented software engineering (AOSE) [6, 10] has also led to the proposal of several models, methodologies and tools to guide the analysis and design of complex systems in this context.

AGENTS IN BIOINFORMATICS

Agent technology deals with entities typically equipped with information management and coordination capabilities. The notion of agents in bioinformatics thus suggests the support of integration of information by designing domain-aware information agents for knowledge management and problem-solving within a biological domain. The use of agents in computational and systems biology suggest the design of agent-based systems, tools and languages for modelling the biological processes themselves.

At the dawn of the 'omics' age, bioinformatics was defined [11] as a computational discipline aiming at the management and analysis of biological data. Nowadays, we should also include in this definition the capability to address information and knowledge overflow as well—integration has become the password. Computational biology focuses more on the algorithmic aspects, often taking into account biomolecular concepts or even mimicking them [12]. Systems biology attempts to understand the emerging behaviour of biological systems as a whole [13]. The three disciplines are so strongly correlated and integrated that in the rest of the article we dwell on them.

Agents proved to be useful for the applications that imply: repetitive and time-consuming activities; knowledge management, such as integration of different knowledge sources and modelling of complex, dynamic systems. All of these are typical tasks in the aforementioned disciplines. In particular, the kinds of resources available in the bioinformatics domain, with numerous databases and analysis tools independently administered in geographically distinct locations, lend themselves almost ideally to the adoption of a multi-agent approach. Here, the environment is open, distributed and dynamic, with resources entering and leaving the system over time. There are likely to be large numbers of interactions between entities for various purposes, and the need for automation is substantial and pressing. Some early work in this direction, using agents for genome analysis, is demonstrated by the GeneWeaver project in the UK [14], and work using DECAF in the US [15, 16]. Earlier work [1, 17] does not mention agents explicitly but shares many similar concepts.

Recently, Keele and Wray [18] reviewed the issues concerning applications of software agent technology to genomics. In Italy, results in the application of agents to data and tools integration, have been provided by the BioAgent project [19–22]. For biological systems simulation, early work demonstrates the use of agent technology to model intracellular signalling pathways [23], and in visual tools for cell modelling [24]. More substantial work is now underway on the use of distributed components as part of the UK's ^{my}Grid [25] e-Science project (http://www.mygrid.org.uk), developing a Bioinformatics Grid testbed, which may also merit the application of the agent paradigm [26]. Another project, with a special applications for biology, is the Italian Grid.it (http://www.grid.it); this project aims to provide platforms for high-performance computational grids oriented at scalable virtual organizations. Promising experimental studies on the integration of Grid and agent technology are also being carried out in the framework of a new project, LITBIO (Interactive Laboratory of Bioinformatics Technologies; http://www.litbio.org).

Recent experiences

In this section, we report on some recent experiences in using agents in bioinformatics, and discuss the results obtained in employing them both as assistants for bioinformaticians and as problem solvers for biologists. More details of each application can be found in the corresponding references.

Bioinformatics process automation

In order to illustrate the role of agents in bioinformatics process automation, we consider the experience of GeneWeaver [14], which is a multi-agent system aimed at addressing concerns with the management of data and analysis methods for bioinformatics. It comprises a community of agents that interact with each other, each having a particular role, in an overall effort to automate processes involved in bioinformatics. The system was targeted at genome annotation, but should not really be viewed as satisfying a single need, with each agent being able to deliver its own expertise at solving particular problems. If we consider the kinds of problems that are common in such applications, including filtering and prioritizing information resulting from matched proteins, integrating several distinct analysis programs possibly in sophisticated ways, managing multiple remote sources of data in different formats, and so on, no solution for automation suggests itself quite as much as a multi-agent approach. In fact, this kind of problem is not really novel—it fits what might be considered a standard model of a multi-agent system in a traditional information systems domain with the addition of some extra complications and a different set of data.

Agents in the system can be concerned with management of the primary databases, performing sequence analyses using existing tools, or with storing and presenting resulting information. The important point to note is that the system does not offer new methods for performing these tasks, but organizes existing ones for the most effective and flexible operation.

There are five types of agents present in the GeneWeaver community.

- Broker agents are facilitators rather than points of functionality, needed to register information about other agents in the community.
- Primary database agents are needed to manage remote primary sequence databases and keep the data contained in them up-to-date, and in a format that allows other agents to query that data.
- Non-redundant database agents construct and maintain non-redundant databases from the data managed by other primary database agents in the community.
- Calculation agents encapsulate some pre-existing methods or tools for analysis of sequence data, and attempt to determine the structure or function of a sequence. Whenever possible, they are also responsible for constructing and managing any underlying data that they rely on.
- Genome agents are responsible for managing the genomic information for a particular organism.

Agents for the analysis of polygenic diseases

It is often not clear if a gene is expressed or differentially expressed. It is even more difficult to determine if an observed change is relevant for a disease. Humans are not good in ranking these findings, particularly not for complex diseases, with many contributing factors. A laboratory investigating a particular disease is likely to have both RNA and protein expression data from many different sources. The data are likely to include information from cell cultures, experiments prior, and post, to the

inhibition of a particular set of genes, nucleotide polymorphisms and the same for animal models of the disease from multiple strains. Also of particular interest is the genotyping of animals, from which identifications of chromosomal loci that contribute to the disease may be inferred.

The use of agents for the analysis of polygenic diseases and preliminary results on combining RNA and protein expression levels, genotyping and intergenomics by adopting BioAgent, a programming environment based on mobile middleware [20], are encouraging. Agent technology supports uniform access to local and public data (through a facilitator, i.e. a wrapper of web services or local tools, as implemented by EDITtoTrEMBL and other efforts [1, 17, 27]). Agent technology helps in understanding links between data sources and their association with diseases, providing reasoning over these data to yield a model of the disease in terms of the minimal number of genes/pathways that explain the maximal number of observations of the disease. Agents gather annotation of protein or genomic sequences and establish a consensus, as implemented for information from protein domain databases and trans-membrane protein sequence annotation [28]. In this context, agents are viewed not only as a technical implementation of distributed computing, but also as a manager of different views on the collected data, from which a complete model needs to be inferred.

Many other sources of information are available as web services that agents may provide. These include selection of nucleotide polymorphisms [29], conversion between genetic and physical distances [30] and inter-genomic consensus regions of disease association [31] which, today, are queried only manually and independently.

Agents could also be used to select preferred investigations of particular regions of 2D-gels, e.g. zooming in on gels, mass spectrometry (MS)-identification of spots, and searching for predicted variants. They might also suggest investigation of genes that are not on a microarray chip by intelligently supporting the huge computational effort required, which could benefit from load sharing in the context of grid computing.

myGrid practical experience in tasks automation

One of the key problems facing bioinformaticians is the task of finding the services and the data that they need to perform *in silico* experiments. This task is complex for several reasons. First, the tools are often widely distributed, maintained by many different decentralized groups. Secondly, there are many different tools, performing many different kinds of operation, on many different kinds of data. And, finally, this is further complicated by the lack of formal standards for representing the data of bioinformatics.

It is against this backdrop that the myGrid project operates. The project has built a service-oriented system that enables the publication and composition of tools as services [32], recognizing that service autonomy and heterogeneity are the key challenges in bioinformatics, rather than the requirement for high performance, which was the original focus of computational Grid technologies. While this simplifies some of the difficulties described earlier—it is no longer necessary to 'screen scrape' web pages—it does not address the difficulties of complexity.

This environment is one that seems ripe for the application of agent technology; coping with distribution, decentralization and complexity are some of the biggest perceived advantages of this technology. However, to enable the use of agents, large amounts of knowledge are required in a form that can be processed by the agent. Towards this aim, myGrid has made heavy use of semantic web technologies, focusing on providing descriptions of the services that support the task of discovering and composing services, in a manner that facilitates and supports the work of the bioinformatician [33]. Currently, this work is very much 'user-oriented': the knowledge is provided by the user and the main service discovery agent is the user. However, it is now investigating techniques for making more automated use of this knowledge, in particular *shim services*—those services that align closely related data—enabling the bioinformatician to combine services without having to worry about complexity and reducing the difficulties resulting from the use of the flat files and informal representations [34].

In addition to the task of discovery of tools and data sets, myGrid has attempted to address some of the difficulties in organizing and storing knowledge about the derivation or provenance of data generated by its service-oriented architecture [35]. This is traditionally a difficulty in bioinformatics, where it is often hard to determine what information was used to draw a given conclusion, resulting in databases that are error prone and possibly circular. Again, the focus has been on user interaction with this data,

but using formats that should be computationally accessible to agents.

The experiences of myGrid highlight some of the barriers to the adoption of agent technology. There is a continual tension between the desire for agents to use rich and expressive knowledge, with the complexity of actually obtaining this information. myGrid has generally used simpler technologies and much less expressive representations. This reduces the effort required to obtain the knowledge, but, probably, also reduces the application of it. Despite these difficulties, myGrid is an evidence of the importance of marrying computer science research with bioinformatics. Semantic web and agent technologies offer much for reducing the complexity of the tasks of bioinformatics, while bioinformatics offers a rich domain with real world problems for the computer scientist. As bioinformatics continues to increase the formalization in the data and the desire for automation, both the resources needed and the requirements for multi-agent systems are becoming clearer.

An agent-based semantic web for bioinformatics

The power of ontologies and the idea of the semantic web is evident from novel applications such as GoPubMed (www.gopubmed.org), an ontology-based literature search engine [36]. In a first step, GoPubMed automatically identifies GeneOntology [37] terms in PubMed literature abstracts and tags the abstracts accordingly. In this respect, GoPubMed changes web contents to semantic web contents. In a second step, it allows users to explore PubMed search results with the GeneOntology. The categories of the ontology help users quickly to survey and group abstracts according to relevant categories rather than working through a list of papers. And, it allows task automation by providing agents with a large amount of knowledge in a form that can readily be processed.

www.Prova.ws: rule-based java scripting

Semantic web applications such as GoPubMed integrate ontologies and other data sources such as PubMed. In general, there is therefore a need for bioinformatics system integration specifically supporting reasoning over structured vocabularies. Prova [38], a language for rule-based Java-scripting, aims to address this need. Prova has been used e.g. to implement the first GoPubMed prototype. The use of rules allows one to specify declaratively the integration needs at a high-level without any implementation details. The transparent integration of Java caters for easy access and integration of database access, web services and many other Java services. This way, Prova combines the advantages of rule-based programming and object-oriented programming in Java. The Prova language is positioned as a platform for knowledge-intensive ontology-rich (most likely, agent-based) applications in biomedical research. It aims to satisfy the following design goals: combine the benefits of declarative and object-oriented programming; merge the syntaxes of Prolog as a rule-based language and Java as an object-oriented language; expose logic as rules; access data sources via wrappers written in Java or command-line shells like Perl; make all Java Application Programming Interfaces (APIs) from available packages directly accessible from rules; run within the Java runtime environment; be compatible with web-based and agent-based software architectures and provide functionality necessary for rapid application prototyping and low-cost maintenance.

Differently from other reasoners (e.g. RACER [39]), Prova supports the use of agents for reasoning over such ontologies and integrating them with databases and web services. Karasavvas and colleagues [40] also argue for the importance of an agent communication language (and a standard derived from it) in the perspective of bioinformatics integration systems. Furthermore, they evaluate criticality issues concerning the decisions to be taken in bioinformatics integration systems [41].

Protein secondary structure prediction

The problem of predicting protein 3D-structure is very complex, as the underlying process involves biological, chemical and physical interactions. A simplified task is to predict the secondary structure, i.e. the local conformation of the peptide chain projected into a one-dimensional sequence. Despite this simplification, information about secondary structure often provides useful information for predicting protein functional sites, which justifies the interest of researchers in this particular and exciting field. Artificial neural networks (ANNs) have been widely applied to this task [42, 43] and represent the core of many successful secondary structure prediction methods, thanks to their ability to find patterns without the need for predetermined models or known mechanisms. In fact, all modern methods actually resort to ensembles of ANNs, usually organized into different functional levels.

In relation to agents, one architecture in which the existence of separate 'experts' is clearly articulated has been proposed in [44]. To predict the secondary structure of a protein, the corresponding system, called *MASSP* (MultiAgent Secondary Structure Predictor), resorts to a population of homogeneous experts—each expert being implemented by a software agent that embodies a genetic and a neural component (i.e. guard and embedded predictor, respectively). Guards and predictors perform different tasks and are supplied with different information. In particular, a guard is aimed at (soft-)partitioning the input space, insomuch as assuring both the diversity and the specialization of the corresponding embedded predictor, which in turn is devoted to perform the actual prediction. Guards deal with inputs that encode information strictly related with relevant domain knowledge, whereas embedded predictors process other relevant inputs, each consisting of a limited window of residues. In the current release of the system, agent technology in its full potential is used, Jade [45] being adopted as underlying programming framework. Although, experimental results are already promising—an accuracy of about 76%, measured in terms of Q_3, has been reached—the adoption of the agent technology is mainly due to the requirements imposed on the next release of the system—which is expected (i) to implement complex interactions, (ii) to implement heterogeneous experts and (iii) to integrate predictions performed by other predictors disseminated over the Internet. In fact, software agents are perfectly suited to fulfil the requirements above, as they offer a new paradigm for very large-scale distributed heterogeneous applications, focused on the interactions of autonomous, cooperating processes (for further details see, for instance, Bradshaw [46]). Regarding the first of the above issues, let us stress that the environment in which MASSP experts operate basically stems from that dictated by the basic rules of evolutionary computation, in which the main schema of interaction is based on competition. Thus, more complex and flexible forms of interaction may be difficult to implement, in particular to enable experts to apply different policies in accordance with the current state of the computation and with the current operational context. Fortunately, interaction is a key focus of agent technology (see, for instance [47]), which involves communication languages and interaction protocols. As for the second issue (i.e. heterogeneity), it is clear that the ability to deal with experts able to process different kinds of data, either locally available, or downloaded from the Internet, creates a scenario in which automated experts can mimic the workflow activity performed by human experts, which are able to cooperate in predicting secondary structures despite the fact that their 'expertise' may derive from different bodies of domain knowledge. Due to their capability of exchanging information, despite their heterogeneity, software agents appear to be the most suitable technology able to deal with this kind of problem. In relation to the third issue (i.e. openness), there is a growing amount of evidence that consensus methods may outperform the accuracy of single predictors [48]. Although MASSP has been designed and implemented to exploit this phenomenon on a local basis, nothing prevents the extension of this approach in such a way that remote predictors may become part of the overall population of experts. This is relatively easy to do by resorting to software agents, as they are also particularly well-suited to acting as wrappers, each hiding the details of the corresponding remote predictor while interacting with other experts involved in the prediction activity.

Stem cell analysis and simulation

In recent years, there has been a growing debate about how stem cells behave in the human body; whether the fate of stem cells is pre-determined or stochastic, and whether the fate of cells relies on their internal state, or on extra-cellular micro-environmental factors. More recent experimental evidence has suggested that stem cell development may be more complicated than was originally thought [49]. New theories challenged the prevailing view suggesting that stem cell fate is both *reversible* (cells can become less differentiated or behave more like stem cells) and *plastic* (cells can migrate from one cell lineage to another). More recently, there has been a growing body of work that is concerned with building predictive formal models of stem cell behaviour that can be simulated. In this direction, much work has been done in building agent-based simulations of stem cells [50–53].

Work to date has used existing, well-established techniques for specifying and modelling agent-based systems in general [54] and progressed along two parallel strands. The first strand has been an attempt to develop an agent-based model of Theise's theory of stem cell behaviour and organization [53].

The second strand has been to use the same agent-based approach to analyse and re-develop existing models to ensure that the agent framework is sufficiently flexible to model more than one theory and to understand how other work differs.

Two of the most common approaches to formal models of stem cells use cellular automata [55] and equational or probabilistic models [56]. In order to support the claim that the agent approach is more suitable than other modelling approaches, existing approaches have been taken and re-cast in the agent-based modelling and simulation framework, which has demonstrated a number of clear advantages of the agent approach over existing approaches [57]. Specifically, the agent model has more biological plausibility, and is thus appropriate as a computer modelling metaphor for interdisciplinary collaboration between modellers and wet lab experimentalists. For example: in the CA models, cells magically appear; in the probabilistic-based models cells have access to global system information; and in the differential-based models we cannot begin to investigate how individual cell–cell interaction leads to the well-documented global system behaviour of cell systems. This is not to say that agents are in any way better than other approaches in general; each has its own merits, of course. It is simply that in this context the agent-based approach has, to date, demonstrated a clear number of advantages.

Furthermore, arguably the most sophisticated current equational-based model of stem cell activity, has been re-caged in an agent framework, demonstrating a number of clear advantages. First, it shows how the environment may limit the behaviour of cells. For example, division is not necessarily guaranteed if there is insufficient space. The agent-based simulation increases the biological intuition and plausibility, and allows the investigation of behaviours due to subtle changes in micro-environmental effects for each cell. This was not possible before. Modelling cells as agents responding autonomously to their local environment is much more fine-grained than using an equational/probabilistic-function approach to model cell transitions, and therefore allows for a much greater degree of sophistication in the possibilities of understanding how self-organization actually takes place in the adult human body.

In this view, the agent approach is more biologically plausible since it does not rely on getting information about the overall system state, and instead its behaviour is based solely on its internal state, its perception of the local environment state, and the actual physical state of the local environment. Biological plausibility at this abstract modelling level is important to attract biologists to use and work with models and simulations in general. Stem cells are a prime example of a self-organizing system where individual agents react to their local physical, chemical and biological environment.

To date, we have produced formal and mutually consistent specifications of the leading of many of the key predictive models of stem cell behaviour within our agent framework. In addition, we have produced simulations and visualizations of these models. And having worked in this field now for around 3 years, it is our belief that visualization of stem cell simulations may hold the key for the integration of new models of stem cell organization into the wet lab culture.

Moreover, using the application of our agent framework we have introduced more biological plausibility to the models (cells as agents is a natural and engaging metaphor for biologists), we have introduced cell mechanisms in place of statistical or probabilistic methods that rely on information about the entire cell population being instantaneously available to all cells, we have produced visualizations that enable a dialogue between wet lab researchers, and we have made predictions about stem cell behaviour that can be investigated in the wet lab. For example, according to our models, stem cell activity pulses around the stem cell niche. We are currently in negotiation with stem cell laboratories to develop an experiment to test our hypothesis relating to this system behaviour (predictions about individual cells cannot be tested in the human body) and thus the corresponding model on which it is based. We are aware of course, that our model is incredibly simple compared with the sophistication of the human body. Nevertheless, we are increasingly confident that the theoretical simplifications inherent in any model will provide crucial understandings into cell interaction mechanisms, and that the agent metaphor provides exactly the right metaphor for continued interdisciplinary collaboration between biologists and the developers of predictive models.

FUTURE CHALLENGES

In this section, we report on several areas for which agents appear to offer a promising technology in support of a new approach.

Analysis of mutant proteins: an exercise in motivation

A potential application of agents could be the problem of collecting data on mutations and analysing their effects on protein structure. Many diseases are caused by DNA mutations which lead to protein mutations: cystic fibrosis, Favism (G6PD), Niemann Picks disease, OTC deficiency (urea cycle—hyper-ammonaemia—brain damage), Cancer (p53, BRCA-1, APC, MYH). Often, biologists who study protein mutations attempt to analyse the protein structure, since structure determines function. Could agents in some way help to provide an answer to the problem of verifying SNPs and confirming whether they are coding, leading to a protein mutation [58]? If so, where is the mutation in the protein sequence and is there a structure already known for such proteins? How does the mutation affect the structure? We could encode a workflow to describe the possible answers to these questions.

The automation of the workflow implies middleware suitable for supporting the specification, execution and coordination of very complex activities. The use of information agents, in the context of the semantic web, could help significantly in retrieving and integrating meaningful information from heterogeneous and distributed data repositories. In collecting information from diverse sources, however, technology is often not the problem: ontologies, web services and agent-based systems are all well-established [59]. Rather, the problem can be in persuading the biologist to agree to use ontologies (The Open Biomedical Ontologies website lists all the available ontologies in the biomedical field http://obo.sourceforge.net/) and nomenclatures [60]. If the technologies are too complex, or perceived to be too complex, then why should the biologist bother? They need to see a direct benefit in making use of such systems. Clearly, if one technology is obviously better than another, then there will be no hesitation in its adoption. However, what may be 'better' for the community as a whole may not be of direct benefit to an individual biologist. In addition, the technology may well be so outside the scope of expertise of a bench biologist that he or she has no concept of how and why it may be useful. Thus, the problem is one of motivation—persuading the biologist who may have collected some interesting data and put it up on the web (e.g. one of the several hundred websites listing mutations for specific proteins [61]), to adopt standards and ontologies [62] that can be used by agents and the semantic web. Thus, to be successful, biologists, bioinformaticians and computer scientists must work closely, but most importantly, must be driven by the needs of the biologist.

LIMS as an agent-based laboratory

An area that would certainly benefit from the agent paradigm is that of Laboratory Information Management Systems (LIMS). More than 160 packages and programs [63] are available for laboratory automation necessities. Most of them represent commercial products, provided from hardware vendors, specifically designed for their laboratory machinery and solutions. The enormous amount of data and metadata [64] produced from high-throughput technologies and projects in the plethora of '-omics' fields (e.g. genome sequencing, microarrays and transcriptomics, proteomics) and in a number of others (e.g. immunofluorescence imaging, flow cytometry, chemical analysis, environmental sciences) require such an information management framework. Very little academic research has been performed on this topic, mainly because of its very strict connection to dedicated equipment and to laboratory-specific data format, requirements and procedures. Without aiming to be exhaustive, we can cite some *ad-hoc* academic solutions to specific problems: QuickLIMS [65] developed for microarrays production, MMP-LIMS [66] used for integrated genetic and physical map in the maize genome project, CLIMS [67] for a crystallography laboratory and the LIMS setup for building the *Pseudomonas aeruginosa* gene collection [68]. LabBase [69] represented a general-purpose database management system for implementation of laboratory information systems. Based on a community-agreed data model and already looking in the e-Science dimension is the MOLE project [63] aiming to serve protein production laboratories in the UK and Europe.

The requirements for a system that would be flexible, scalable and capable to easily adapt to any change, without engendering any traumatic event for the laboratory [70] are evident. It should also be noted that until recently automation has focused primarily on improving hardware. Future advances will concentrate on intelligent software to integrate physical experimentation and result analysis with hypothesis formulation and experiment planning [64, 71]. We argue that the agent metaphor, integrated with appropriate and detailed domain

ontologies, could intuitively describe and manage distributed environments populated by autonomous entities that wrap robotized stations, interface human operators, describe laboratory objects (e.g. samples, well, plates), operations and procedures. Intelligent agents would also be capable of successfully coping with fast-changing (due to the ever increasing technological turnover) and unpredictable conditions.

Cellular processes modelling

The modelling of cellular processes is difficult due to the complexity of the organization of biological systems and of its cellular processes. Modelling complex systems implies a deep understanding of the system both in terms of its structure and its behaviour [13]. Once we have identified some of the components, some of their functions, their topological relationships and the parameters of each relation, we can start to analyse the system behaviour trying to understand the mechanisms behind the robustness and stability of the system. At present, the unavailability of complete knowledge, leads to an unavoidable degree of uncertainty in our models. To this end, agent technology can be exploited to develop a suitable conceptual framework for simulation in order to analyse system behaviour and eventually to infer new components and functions. One proposed exercise is to analyse the cell in terms of the known active components, the roles and behaviours these play in the cell processes, their interactions with the living environment. But, in approaching the agent-based cell simulation, at what abstraction level should we model the cell system—the fine grained level? What would be the main features of an agent-based conceptual framework for simulation of biological systems? What would be a good bioagent conceptual language?

Based on the consideration that biological systems are complex, consisting of a set of components interacting with each other and with an external (dynamic) environment, a conceptual framework for engineering an agent society that simulates the behaviour of a biological system has been proposed [72]. In contrast to the classical mathematical descriptions mainly based on ordinary differential equations, the specification of complex systems is based on behavioural modelling. For example, an agent-based model of the carbohydrate oxidation in the cell, describing each engineering step by Unified Modelling Language (UML) graphical notation has already been suggested [73]. Other recent examples of application of agent technology in systems biology concern the tissue homeostasis in human epidermis [74], bacterial chemotaxis [75], molecular self-organization [76] and T-cell recognition [77]. Other approaches, not agent-based, are relevant in the cell modelling and simulation context: Cell-DEVS [78], based on discrete-events systems specification, E-CELL [79], for modelling biochemical and genetic processes, Virtual Cell [80], a general framework for the spatial modelling and simulation of cellular physiology and Physione [81], for modelling human body from a fine to coarse grain level.

Formal and semiformal methods in bioinformatics

In addition to the expected contribution of agents in bioinformatics as a technological framework, we see another challenge to deal with, i.e. the possibility of designing *incredibly* complex systems, through models suitable for representing and analysing biological systems from different viewpoints: static-structural, dynamic and functional [73, 82]. In fact, the use of models to represent a biological system at different abstraction levels helps us to understand the biological system itself. The specification model, e.g. agent-oriented, can help by identifying the system structure, critical component roles and responsibilities, functions and interactions (which are generally poorly identified). Of course, to create models we need languages and suitable notations.

In the literature, a wide range of formal and semi-formal languages and notations can be found. These depend on the level considered, on the properties in which the designer is interested, and on the tools available to perform the analysis and verify properties. Proving properties in biological models can mean verifying properties related to the system/process behaviour (e.g. safety properties; liveness properties; simulations of system dynamics; checking for causal relationships ...). Any property can be formally proved by using well-known methods such as equivalence checking, model checking, simulation and model synthesis.

In a very simple scenario, a semi-formal notation, based on PetriNets [82] and UML Activity Diagrams [83], is used to graphically describe the workflow activities for a biological process. In particular, Figure 1 shows four different models to represent the *malaria parasite invading human host erythrocytes* system at different levels of abstraction with bold

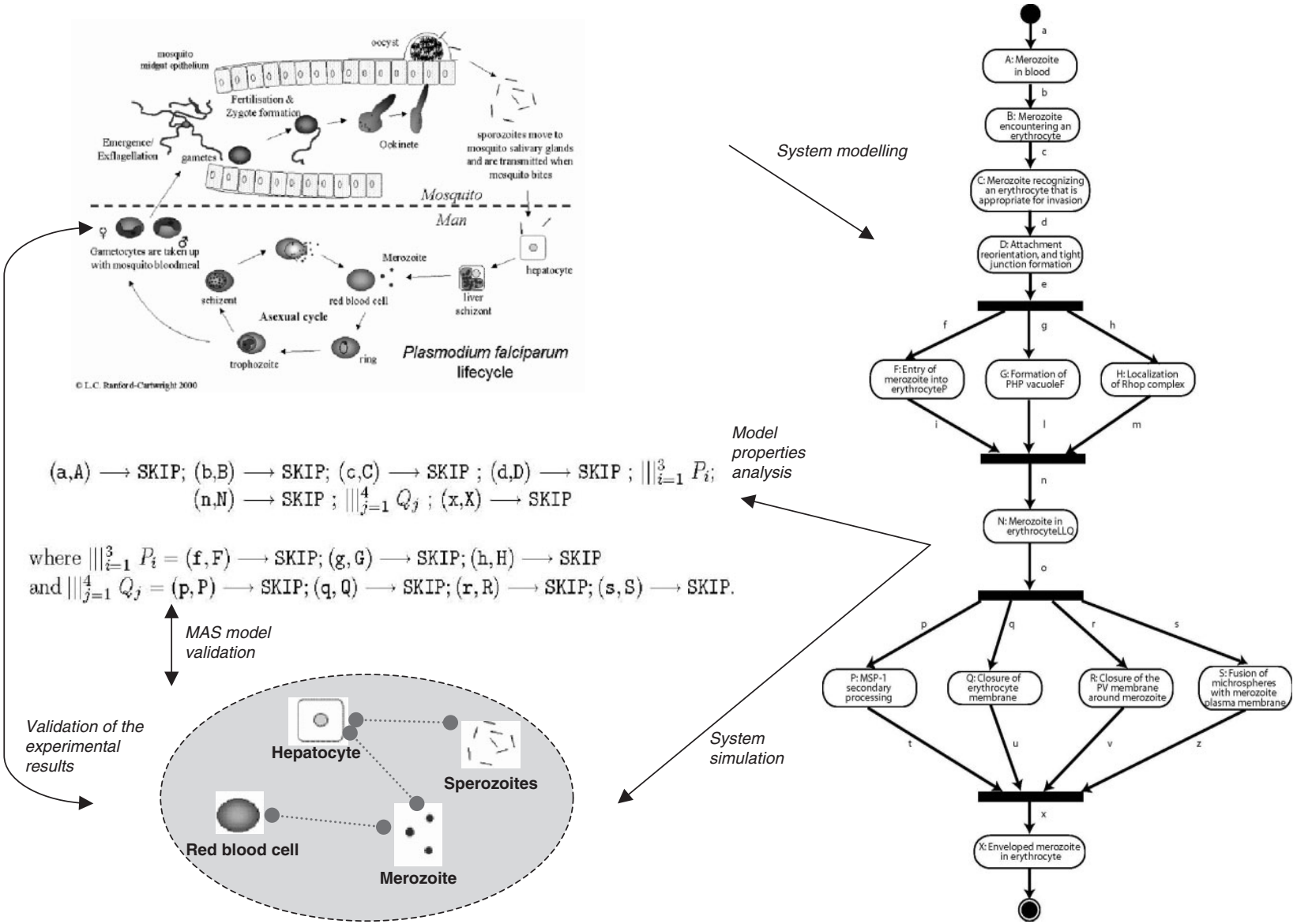

Figure I: Engineering bioinformatics: from biological model to multi-agent system. A biological process can be described by a graphical semi-formal model, validated by a formal model, simulated by a multi-agent system and finally tested with the experimental results.

arrows indicating the steps (modelling, analysing, simulation and validation) to derive one model from another.

Starting form the biological knowledge of a system, a graphical (semi-formal) description can be derived. Then on the one hand, this model is translated to a formal specification (by process algebra) to verify the model's properties, and on the other hand, the graphical description can be compiled into a low-level specification (in an agent-oriented language) to generate the agent-based simulation of the biological system. The last step, the software validation [84] of the multi-agent system [85], can give rise to an enrichment of the formal model by including properties to make the model more faithful to the biological system [86]. The natural question, therefore, is how we can know what kind of system properties biologists want to verify. Are they interested in having clear evidence of how the simulation system behaves, being able to modify the system's properties at run-time? And should it be possible to incrementally build, maintain and refine the system? What kind of conceptual simulation framework would be useful to fulfil biologists' expectations? Would an agent approach be sufficient to create a framework with these features [72]? Would mobility be a meaningful feature to simulate biological systems through agent technology? (Note that in Figure 1 some system components are mobile.)

CONCLUSION

It is clear that the combination of agents and bioinformatics presents a 2-fold opportunity. On the one hand, the domain of bioinformatics, with its extensive and growing resources of databases and analysis tools, provides an appropriate domain for the application of agent technologies. It offers the possibility for deploying and testing agent systems in a real-world setting with the possibility of making substantial contributions to human society. On the other hand, there is a distinct and identified need for good solutions to improve the performance of existing bioinformatics systems, and agents may be able to contribute to that improvement. In this sense, there is a very strong synergy between the two domains.

This picture is both enhanced and complicated by the introduction of relevant infrastructural technologies that facilitate both bioinformatics and agent-based computing. For example, the Grid has become increasingly important to both the communities, and suggests a convergence to a service-oriented vision of bioinformatics underpinned by Grid-based virtual organizations.

However, there are still significant challenges. Researchers from both communities generally require education in the other, and work must be undertaken to ensure that any solutions across both areas satisfy both needs. In many cases, the language of discourse is so distinct that discussion of key issues becomes problematic. Additionally, the introduction of new technologies like the Grid requires further efforts, both in terms of understanding and adoption, and in terms of its immaturity in fully deployed systems. Maturity at the interface is thus the key challenge. While many agent techniques may be used to address the concerns of bioinformaticians, the lack of a complete understanding across domains suggests that it may still be too early to develop more sophisticated systems than the current generation of essential management and mediation systems.

Key Points

- Agents provide designers and developers with a way of structuring an application around autonomous, communicative elements.
- The combination of agents and bioinformatics presents a 2-fold opportunity.
- An agent is a computer system capable of flexible, autonomous problem-solving actions.
- The use of information agents, in the context of the semantic web, could help significantly in retrieving and integrating meaningful information from heterogeneous and distributed data repositories.
- At what abstraction level should we model the cell system?
- Agents can be considered as a distinct kind of software abstraction.
- What would be a good bioagent conceptual language?
- Would mobility be a meaningful feature to simulate biological systems through agent technology?

Acknowledgements

The work was supported by the grant N. 03-51-5218 INTAS, MIUR FIRB Bioinformatics and LITBIO projects.

References

1. Gaasterland T, Sensen C. Fully automated genome analysis that reflects user needs and preferences. A detailed introduction to the magpie system architecture. *Biochimie* 1996;**78**:302–10.
2. Fleischmann W, Möller S, Gateau A, Apweiler R. A novel method for automatic functional annotation of proteins. *Bioinformatics* 1999;**15**:228–33.

3. Luck M, McBurney P, Preist C. Agent technology: enabling next generation computing (a roadmap for agent based computing). Technical report, AgentLink II, 2003.
4. Luck M, Merelli E. Agents in bioinformatics. *KER* 2005;**20**: 117–25.
5. Jennings N. An agent-based approach for building complex software systems. *Communications of the ACM* 2001;**44**:35–41.
6. Zambonelli F, Omicini A. Challenges and research directions in agent-oriented software engineering. *Autonomous Agents and Multi-Agent Systems* 2004;**9**:253–83.
7. Wooldridge M, Jennings N. Agent theories, architectures and languages: a survey. In: *Intelligent Agents, ECAI-94 Workshop on Agent Theories, Architectures and Languages.* Amsterdam, The Netherlands: Springer-Verlag, 1994: pp. 1–39.
8. Jennings N, Wooldridge M. Applications of intelligent agents. In: *Agent Technology: Foundations, Applications, and Markets.* New York: Springer-Verlag, 1998.
9. Finin T, Fritzson R, McKay D, McEntire R. KQML as an Agent Communication Language. In: *Proceedings of the 3rd International Conference on Information and Knowledge Management*. Maryland, United States: ACM Press, 1994: pp. 456–63.
10. Gervais M, Gomez J, Weiss G. A survey on agent oriented software engineering reseach. In: *Methodologies and software engineering for agent systems*. New York: Kluwer, 2004.
11. Boguski M. Bioinformatics – a new era. *TIBS* 1998;**19**:1–3.
12. Claverie J-M. From Bioinformatics to Computational Biology. *Genome Res* 2000;**10**:1277–9.
13. Kitano H. *Foundations of Systems Biology.* Cambridge, Massachussetts: MIT Press, 2002.
14. Bryson K, Luck M, Joy M, Jones D. Applying agents to bioinformatics in Geneweaver. In: *Cooperative Information Agents IV, Lecture Notes in Artificial Intelligence*. Berlin/ Heidelberg: Springer-Verlag, 2000: pp. 60–71.
15. Graham JRA, Decker K, Mersic M. Decaf – a flexible multi agent system architecture. *Auton Agents Multi Agent Syst* 2003;**7**:7–27.
16. Decker K, Khan S, Schmidt C, *et al*. Biomas: a multi-agent system for genomic annotation. *Int J Cooperative Inf Systems* 2002;**11**:265–92.
17. Möller S, Leser U, Fleischmann W, Apweiler R. EDITtoTrEMBL: a distributed approach to high-quality automated protein sequence annotation. *Bioinformatics* 1999; **15**:219–27.
18. Keele JW, Wray J. Software agents in molecular computational biology. *Brief Bioinform* 2005;**6**:370–9.
19. Angeletti M, Culmone R, Merelli E. An intelligent agent architecture for DNA-microarray data integration. In *NETTAB — CORBA and XML: Towards a bioinformatics integrated network environment*, Genova, Italy, 2001.
20. Merelli E, Culmone R, Mariani L. Bioagent: A mobile agent system for bioscientists. In: *NETTAB — Agents in Bioinformatics*, Bologna, Italy, 2002: pp. 99–100.
21. Corradini F, Mariani L, Merelli E. An agent-based approach to tool integration. *J Software Tools Technology Transfer* 2004;**6**: 231–44.
22. Corradini F, Merelli E. Hermes: agent-based middleware for mobile computing. In: *Tutorial Book of SFM-05*. Springer-Verlag, 2005, LNCS 3465.
23. Gonzalez P, Cardenas M, Camacho D, *et al*. Cellulat: an agent-based intracellular signalling model. *BioSystems* 2003; **68**:171–85.
24. Webb K, White T. Cell modeling using agent-based formalisms. *Auton Agent Multi Agent Syst* 2004.
25. Stevens RD, Robinson AJ, Goble CA. myGrid: personalised bioinformatics on the information grid. *Bioinformatics* 2003;**19**(Suppl 1):i302–4.
26. Moreau L, Miles S, Goble *et al*. On the use of agents in a bioinformatics grid. In: *NETTAB — Agents in Bioinformatics*, Bologna, Italy, 2002.
27. Bartocci E, Mariani L, Merelli E. An XML view of the 'world'. *International Conference on Enterprise Information Systems*, Angers, France 2003;19–27.
28. Möller S, Schroeder M, Schroeder M. Conflict-resolution for the automated annotation of transmembrane proteins. *Computational Chemistry* 2001;**26**:41–6.
29. Möller S, Koczan D, Serrano-Fernandez P, *et al*. Selecting SNPs for association studies based on population frequencies: a novel interactive tool and its application to polygenic diseases. *Silico Biology* 2004;**4**:417–27.
30. Voigt C, Möller S, Ibrahim SM, Serrano-Fernandez P. Non-linear conversion between genetic and physical chromosomal distances. *Bioinformatics* 2004;**20**:1966–7.
31. Serrano-Fernandez P, Ibrahim MS, Zettl UK, *et al*. Intergenomic consensus in multifactorial inheritance loci: the case of multiple sclerosis. *Genes Immun* 2004;**5**:615–20.
32. Oinn T, Addis M, Ferris J, *et al*. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 2004;**20**:3045–54.
33. Lord P, Bechhofer S, Wilkinson MD, *et al*. Applying semantic web services to bioinformatics: experiences gained, lessons learnt. In: *International Semantic Web Conference ISWC*. Heidelberg: Springer-Verlag, 2004: pp. 350–64.
34. Hull D, Stevens R, Lord P, Wroe C, Carole Goble. Treating 'shimantic web' syndrome with ontologies. In: *Workshop proceedings CEUR-WS.org*. UK. ISSN:1613-0073, 8 December 2004.
35. Zhao J, Wroe C, Goble C, *et al*. Using Semantic Web technologies for representing e-Science provenance. In: *Proceedings of Third International Semantic Web Conference (ISWC2004), Hiroshima, Japan, November 2004*. Berlin/ Heidelberg: Springer-Verlag, 2004: pp. 92–106, LNCS 3298.
36. Doms A, Schroeder M. GoPubMed: Exploring PubMed with the GeneOntology. *Nucleic Acid Res* 2005; **33**:W783–86.
37. Ashburner M, Ball CA, Blake JA, *et al*. Gene ontology: tool for the unification of biology. *Nat Genet* 2000;**25**:25–9.
38. Kozlenkov A, Schroeder M. PROVA: rule-based Java-scripting for a bioinformatics semantic web. In: Rahm E, (ed). *International Workshop on Data Integration in the Life Sciences DILS*. Leipzig, Germany: Springer, 2004.
39. Haarslev V, Moller R. RACER System Description. *Lecture Notes In Computer Science* 2001;**2083**:701–6.
40. Karasavvas KA, Baldock R, Burger A. Bioinformatics integration and agent technology. *J Biom Infor* 2004;**37**: 205–19.
41. Karasavvas AK, Baldock R, Burger A. A criticality-based framework for task composition in multi-agent bioinformatics integration systems. *Bioinformatics* 2005;**21**: 3155–63.

42. Hirst JD, Sternberg MJE. Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks. *Biochemistry* 1992;**31**:7211–18.
43. Chandonia JM, Karplus M. Neural networks for secondary structure and structural class prediction. *Protein Science* 1995; **4**:275–85.
44. Armano G, Mancosu G, Orro A. A multi-agent system for protein secondary structure prediction. In: *NETTAB — Models and Metaphors from Biology to Bioinformatics Tools*. Italy: Camerino, 2004: pp. 19–29, 5–7 September 2004.
45. Bellifemine F, Poggi A, Rimassa G. Jade — a FIPA2000 compliant agent development environment. In: *Proceedings of the Fifth International Conference on Autonomous Agents*. Montreal, Canada, 2001.
46. Bradshaw JM. An introduction to software agents. In: Bradshaw JM, (ed). *Software Agents*. Cambridge, MA: AAAI Press/The MIT Press, 1997: pp. 3–46, Chapter 1.
47. Shoham Y. Agent-oriented programming. *Artif Intell* 1992; **60**:51–92.
48. Albrecht M, Tosatto SCE, Lengauer T, Valle G. Simple consensus procedures are effective and sufficient in secondary structure prediction. *Protein Eng* 2003;**16**:459–62.
49. Theise ND, Krause DS. Toward a new paradigm of cell plasticity. *Leukemia* 2003;**16**:542–8.
50. d'Inverno M, Saunders R. Agent-based modelling of stem cell organisation in a niche. In: *Engineering Self-Organising Systems*, volume 3464 of *LNAI*. Berlin/Heidelberg: Springer, 2005.
51. d'Inverno M, Theise ND, Prophet J. Mathematical modelling of stem cells: a complexity primer for the stem cell biologist. In: Potten Christopher, Watson Jim, Clarke Robert, Renehan Andrew, (eds). *Tissue Stem Cells: Biology and Applications*. Marcel Dekker, 2005.
52. Prophet J, d'Inverno M. Transdisciplinary research in cell. In: Paul Fishwick, (ed). *Aesthetic Computing*, Cambridge, USA: MIT Press, 2006, ISBN: 0-262-06250-X.
53. Theise ND, d'Inverno M. Understanding cell lineages as complex adaptive systems. *Blood Cells Mol Dis* 2003;**32**: 17–20.
54. d'Inverno M, Luck M. *Understanding Agent Systems*. 2nd edn. Berlin/Heidelberg/NewYork: Springer, 2003.
55. Agur Z, Daniel Y, Ginosar Y. The universal properties of stem cells as pinpointed by a simple discrete model. *Mathematical Biology* 2003;**44**:79–86.
56. Roeder I. *Dynamical modelling of hematopoietic stem cell organisation*, PhD Dissertation, Leipzig University, 2003.
57. d'Inverno M, Prophet J. Modelling, simulation and visualisation of adult stem cells. In: Gonzalez P, Merelli E, Omicini A, (eds). *Fourth International Workshop on Network Tools and Applications, NETTAB* 2004: pp. 105–16.
58. Cavallo A, Martin ACR. Mapping SNPs to protein sequence and structure data. *Bioinformatics* 2005;**21**:1443–50.
59. Martin ACR. Can we integrate bioinformatics data on the Internet? *Trends Biotechnol* 2001;**19**:327–8.
60. den Dunnen JT, Antonarakis SE. Mutation nomenclature extensions and suggestions to describe complex mutations: A discussion. *Hum Mutat* 2000;**15**:7–12.
61. Claustres M, Horaitis O, Vanevski M, Cotton RGH. Time for a unified system of mutation description and reporting: a review of locus specific mutation databases. *Genome Res* 2002;**12**:680–8.
62. Lazebnik Y. Can a biologist fix a radio? — Or, what I learned while studying apoptosis. *Cancer Cell* 2002; **2**:179–82.
63. Morris C, Wood P, Griffiths SL, *et al*. MOLE: a data management application based on a protein production data model. *Proteins* 2005;**58**:285–9.
64. Whelan KE, King RD. Intelligent software for laboratory automation. *Trends Biotechnol* 2004;**22**:440–5.
65. Kokocinski F, Wrobel G, Hahn M, Lichter P. QuickLIMS: facilitating the data management for DNA-microarray fabrication. *Bioinformatics* 2003;**19**:283–4.
66. Sanchez-Villeda H, Schroeder S, Polacco M, *et al*. Development of an integrated laboratory information management system for the maize mapping project. *Bioinformatics* 2003;**19**:2022–30.
67. Fulton K, Ervine S, Faux NG, *et al*. CLIMS: crystallography laboratory information management system. *Acta Crystallogr D Biol Crystallogr* 2004;**60**:1691–3.
68. LaBaer J, Qiu QQ, Anumanthan A, *et al*. The *Pseudomonas aeruginosa* PA01 gene collection. *Genome Res* 2004;**14**: 2190–200.
69. Goodman N, Rozen S, Stein LD, Smith AG. The LabBase system for data management in large scale biology research laboratories. *Bioinformatics* 1998;**14**:562–74.
70. Avery G, McGee C, Falk S. Implementing LIMS: a 'how-to' guide. *Anal Chem* 2000;**72**:57A–62A.
71. King RD, Whelan KE, Jones FM, *et al*. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* 2004;**427**:247–52.
72. Cannata N, Corradini F, Merelli E, *et al*. An agent-oriented conceptual framework for systems biology. *Transactions on Computational Systems Biology III*, LNBI 3737, 2005.
73. Corradini F, Merelli E, Vita M., A multi-agent system for modelling the oxidation of carbohydrate cellular process. In: *Proceedings of the International Conference on Computational Science and its Applications-Modelling Complex Systems*, LNCS 3481. Berlin/Heidelberg: Springer Verlag, 2005: pp. 1265–73.
74. Grabe N, Neuber K. A multicellular systems biology model predicts epidermal morphology, kinetics and Ca^{2+} flow. *Bioinformatics* 2005;**21**:3541–7.
75. Emonet T, Macal CM, North MJ, *et al*. AgentCell: a digital single-cell assay for bacterial chemotaxis. *Bioinformatics* 2005; **21**:2714–21.
76. Troisi A, Wong V, Ratner MA. An agent-based approach for modeling molecular self-organization. *PNAS* 2005;**102**: 255–60.
77. Casal A, Sumen C, Reddy T, *et al*. Agent-based modelling of the context dependency in T cell recognition. *J Theor Biol* 2005;**236**:376–91.
78. Wainer GA, Giambiasi N. Cell-DEVS/GDEVS for Complex Continuous Systems. *Simul-T Soc Mod Sim* 2005;**81**:137–51.
79. Tomita M, Hashimoto K, Takahashi K, *et al*. E-CELL: software environment for whole-cell simulation. *Bioinformatics* 1999;**15**:72–84.
80. Loew LM, Schaff JC. The Virtual Cell: a software environment for computational cell biology. *Trends Biotechnol* 2001;**19**:401–6.

81. Hunter PJ, Borg TK. Integration from proteins to organs: the Physiome project. *Nat Rev Mol Cell Biol* 2003; **4**:237–43.

82. Peleg M, Yeh I, Altman R. Modeling biological process using workflow and Petri nets. *Bioinfomatics* 2002; **18**:825–37.

83. Amici R, Cacciagrano D, Corradini F, Merelli E. A process algebra view of coordination models with a case study in computational biology. In: *Proceedings of First International Workshop on Coordination and Petri Nets, PNC'04*, Bologna, Italy, 2004.

84. Banks J, Gerstein D, Searles SP. Modeling Processes, Validation, and Verification of Complex Simulations: A Survey, *Methodology and Validation Simulation Series*, Vol. 19(1), The Society for Computer Simulation, 1988, 13–8.

85. Rouff CA, Hinchey M, Rash J, *et al*. Agent Technology from a Formal Perspective. In: *NASA Monographs in Systems and Software Engineering*. Springer-Verlag, 2005.

86. Merelli E, Young M. Validating MAS models with mutations. In: *Proceedings of the First International Workshop on Multi-agent Systems for Medicine, Computational biology and Bioinformatics*. AAMAS, 2005.