An Interdisciplinary Curriculum in Science Informatics

Pat Langley (langley@asu.edu)
School of Computing and Informatics
Arizona State University, Tempe, AZ 85287 USA

Will Bridewell (willb@cslistanford.edu)
Computational Learning Laboratory
Center for the Study of Language and Information
Stanford University, Stanford, CA 94305 USA

Abstract

We are developing a curriculum in science informatics that cuts across disciplinary boundaries to educate students in the computational character of the scientific enterprise. We are designing courses that cover: the storage, retrieval, and analysis of scientific data; the representation of scientific models and their use for prediction and explanation; the forms of communication and interaction that support scientific communities; and the mechanisms that underlie scientific creativity and discovery. We maintain that science has always been an inherently computational endeavor, but that the advent of information technology offers opportunities for aiding or automating every aspect of research. Courses accessible to both undergraduate and graduate students will teach the general concepts and principals that underpin the scientific method, while exploring particular informatics tools in the context of specific problems. Students from different fields will participate in the same courses, which will also serve to encourage interdisciplinary thinking that cuts across domain boundaries. They may follow these with advanced classes that examine in more detail applications in specific scientific fields. In this paper, we motivate the need for a broad curriculum in science informatics and describe the content we are planning to incorporate in our courses.

July 4, 2008

Comments on this document are welcome, but please do not distribute it without permission.

1. A Vision for Science Informatics

In recent years, information technology has changed our lives, altering the manner in which we pursue both personal and professional activities. New digital hardware and software has given us new ways to communicate with each other, learn new facts, listen to music and watch videos, plan trips, shop and purchase products, bank and pay bills, and even give presentations. Computers support all of these activities by storing, retrieving, processing, and interchanging information in digital form. This *informatics revolution* has had major impacts on our culture.

Centuries ago, a much earlier change – the *scientific revolution* – had equally profound effects on society. By collecting data systematically, stating clear theories, and relating them to each other, the scientific method has increased greatly our understanding of the universe, the Earth, matter, life, disease, the mind, and even society itself. The resulting insights about the structures and processes of nature have led in turn to new technologies that have given us abilities that seemed unthinkable to earlier generations.

Clearly, the potential benefits enabled by combining these two revolutionary movements is greater than those made possible by either in isolation. This approach has already produced some impressive advances in a number of scientific fields. Examples have included the collection and analysis of photographic sky surveys, simulation and visualization of fluid dynamics, finding regularities in images from Earth-observing satellites, analysis of genetic sequences and clustering of gene expressions, and the recording/analysis of brain activity. These are each important contributions in their own right, but they only touch on the full potential of combining computer technology with the scientific method.

For example, consider the role that information technology might play in a typical day in the life of a future scientist:

> Professor Jones comes into her office on Tuesday morning. Her first action is to check the status of an experiment she submitted the night before. Her computerized assistant reports the results for culture growth over time under 20 different conditions, displaying each curve in relation to the current model's predictions. The system highlights two conditions in which results diverged from those expected.
>
> Dr. Jones asks the assistant if it has any explanations to propose for the two anomalies, and the system returns a list of ten alternatives, ranked by fits to the data and consistency with knowledge of the field. The scientist asks the computer aide if the literature contains other reports of either similar results or similar hypotheses. It recommends five papers in response; she downloads the two that seem most relevant and she spends the next hour reading them.
>
> After some thought, Dr. Jones tells her assistant to focus on the three model revisions that she feels are most plausible and asks it to design a new experiment that will discriminate among them. The researcher makes a few changes to the assistant's proposed conditions and submits the design for robotic execution when the resources become available. She posts the previous results, the hypothesized explanations, and the new experimental design to her laboratory's site, to keep her colleagues up to date on developments. Dr. Jones leaves the office in time to walk across campus to attend a weekly committee meeting.

We believe that such a future is both possible and desirable, but also that we must take specific steps to achieve it. Some of these involve the development of new information technology, but it is equally important to train scientists in the principles and operation of such informatics tools.

We are not the first to mention the opportunities and challenges in this area. A recent report from the Presidents Information Technology Advisory Committee (2005) stated:

> Universities ... must make coordinated, fundamental, structural changes that affirm the integral role of computational science in addressing the 21st century's most important problems, which are predominantly multidisciplinary ... and collaborative.

To achieve the above vision, we need comprehensive and systematic plans for educating the next generation of scientists in the use of information technology to support their research efforts.

We will refer to this movement as *science informatics*, which we prefer over the more common terms of *e-science* and *computational science*. In its broadest sense, we can define science informatics as *the use of computational metaphors and methods to understand and support the scientific enterprise*. In the next section, we make some claims about this emerging paradigm, some of which differ from views that have appeared in the literature. We then consider implications of these claims for a curriculum in science informatics. After this, we describe the core courses we are developing for such a curriculum at Arizona State University, followed by the outcomes we expect and the manner in which we will assess them. In closing, we consider some broader implications of computational education for scientists.

2. Claims about Science Informatics

Digital computers have been used in science almost since their inception. During that time, a number of distinct communities have emerged, each with its own views on the proper role of information processing in fostering scientific research. Although all of these perspectives have merit, each approaches science informatics from a single, limited angle. In this section, we present three distinctive claims about computing in science that we will use to guide the development of our curriculum.

2.1 Computational Nature of Science

One commonly stated view is that computing and informatics is changing the basic operation of science (PITAC, 2005). The typical formulation is that traditional science has stood on two legs – theory and observation – whereas information technology provides a third leg – computer simulation – that has far-reaching effects. In contrast, we hold that this view is too radical on one front and too conservative on another.

The position is too radical because science has always been an inherently computational endeavor. Human researchers used pencil and paper, blackboards, slide rules, and other tools to make calculate predictions from their models long before the advent of the digital computer, and they engaged in computation in the process.[1] Research in cognitive science and artificial intelligence (Kulkarni

1. Recall that the word *computer* originally referred to a human calculator.

& Simon, 1988; Langley et al., 1987) has shown that we can understand very many aspects of human cognition, including those that arise in scientific research, in computational terms. It seems reasonable to assume that we can model, automate, or assist any facet of science in which humans engage. If so, then information technology will not change the basic character of science. Computational tools may increase researchers' efficiency, reduce their errors, and let them tackle larger problems than before, but the basic steps and relations among them will remain the same.

At the same time, the standard view is too narrow. Placing computational methods in a separate box that contains only a few methods, like numerical simulation and data mining, is overly constraining. Certainly, these are successful technologies with notable success stories, but they undersell the true potential of computational methods. Informatics tools can aid scientists in *every* activity they pursue, from collecting data and explaining results to finding relevant literature and getting their own papers published. The inherently computational nature of science means that digital tools can aid in data collection, model simulation, theory formation, community interaction, and many other activities. Science has always had more than two legs, and informatics offers not a new appendage, but rather provides seven-league boots for those that already exist.

2.2 Breadth of Coverage

Our second claim relates closely to the first. We have argued that information technology is not limited to one facet of science, but rather cuts across the entire range of scientific endeavor. This suggests that, if a discipline engages in some activity, then one can develop computational tools that support it, either by automating it entirely or by providing interactive software that aids human users. The aim of such mechanisms is not to replace scientists entirely, but to give them power tools that let them make progress more rapidly or tackle tasks of size and complexity they could not otherwise handle. From an educational perspective, the main question involves how to organize these tools within a curriculum.

Because many informatics tools for science already exist, there is a natural tendency to organize courses around sets of techniques, but this would miss an opportunity to give students insights into the nature of the scientific method. A more promising approach would start with an analysis of the scientific enterprise into component tasks and then organize the curriculum around them, as we describe at more length in a later section. This would not exclude courses on specific informatics techniques, but students will encounter them at an advanced level, after they have come to understand where they fit within the larger context of science.

2.3 Generality of Principles

As noted earlier, many technical advances in science informatics have occurred in the context of specific disciplines. Algorithms and software for the simulation of differential equation models (Cohen & Hindmarsh, 1996) were initially developed by computational physicists interested in predicting weather, fluid flow, and related phenomena. More recently, notable successes in biology have relied on computational techniques designed for analyzing genomic sequences (Altschul et al.,

1994). Such results have encouraged many to believe that each field should develop its own domain-specific technologies for scientific processing.

In contrast, we believe that science informatics rests on general principles that hold across all disciplines. There are genuine differences in the types and amounts of data available in various fields, as well as distinctions in the forms and use of scientific models. But there also exist common relations and processes that cut across the sciences, and there are clear benefits to developing computational tools that embody them. We should educate researchers in these general principles of the scientific method and their instantiation in the technologies of science informatics.

3. Characteristics of Science Informatics Courses

For the next generation of scientists to be effective, its members must master the principles of science informatics and gain experience with specific technologies that will aid them in their research careers. The goal of educating these scientists interacts with the claims we have just made about science informatics to suggest characteristics of a curriculum, to which we now turn.

3.1 Broad Accessibility

First we should consider the target audience for the core courses. Although students who are pursuing degrees in computer science or informatics would benefit from them, our concern here is with a wider audience. We hope to spread the principles and techniques of science informatics as broadly as possible, especially to students who are seeking degrees in scientific disciplines. In their case, the core courses could contribute to a minor in science informatics that would prepare them to use information technology in their future scientific careers.

This means designing the courses carefully so they do not rely on more knowledge about informatics than science students are likely to have acquired already. Fortunately, most students now come to the university with considerable experience in using information technology, and we can assume they have taken one or two basic courses in the area related to computing. We cannot assume they have sophisticated programming skills, but we can suppose some familiarity with key ideas of computing and informatics that can be taught in an introductory course, which we are also developing for our broader informatics curriculum. Moreover, we can utilize high-level software environments in the science informatics core that are accessible to computing novices.

3.2 Intermediate Level

A related issue concerns the appropriate level at which to offer the core courses. We will assume that participants have already learned some basic content in one or more scientific disciplines, which will help motivate the informatics concepts that are presented. We cannot limit the curriculum to graduate students if we want to have the widest possible impact, yet we do not want to keep graduate students from taking them, since they are unlikely to have had similar training in science informatics in their undergraduate studies at other universities.

This suggests that we offer the core courses at an intermediate level that can be taken both by upper-division undergraduates and by junior graduate students. In this manner, students graduating with either a Bachelors or a Masters degree will have broad preparation for positions in industry or for graduate work in their scientific field. Unfortunately, some universities like ASU insist that undergraduate and graduate courses be distinguished in some manner, but we can address this constraint by incorporating additional requirements for those students who are taking them for graduate credit.

3.3 Content Organization

Another decision involves the organization of the course sequence in terms of content. One approach would be to create courses that reflect traditional topics in science informatics. Examples include numerical analysis for simulation of differential equation models, visualization of scientific data, data mining over scientific data sets, ontologies for scientific data integration, and information retrieval for scientific documents. Many universities already offer a number of such courses. However, these topics would fail to cover the full scope of the scientific enterprise, and they would reinforce views introduced by technologists rather than addressing the needs of scientists.

We believe that a more constructive organization would reflect the major components of the scientific process. Such a sequence would include courses for storage, retrieval, and processing of scientific data, for representation, creation, and use of scientific models, for communication and interaction in scientific communities, and for discovery and creation of new scientific knowledge. Such courses would cut across traditional boundaries and reflect more closely the analyses of science provided by historians, philosopshers, and psychologists. Moreover, specific informatics technologies will change over time, while the basic components of science will remain constant. Section 4 presents a set of courses on these topics in some detail.

3.4 Interdisciplinary Nature

A fourth issue concerns whether courses should focus on informatics for individual scientific fields or attempt to cut across disciplinary boundaries. Traditionally, the term *informatics* has been associated with specific areas of science, as in *medical informatics*, *bioinformatics*, and *ecological informatics*. Undoubtedly, some informatics methods are more relevant to certain fields than others, and there is certainly room for domain-specific courses at more advanced levels.

However, we believe this trend has obscured underlying similarities among the various sciences, and we claimed earlier that there exist principles of scientific representation and processing that have considerable generality. This belief suggests that the core courses should draw their example problems from multiple scientific fields, including the physical, life, and social sciences. Such broad coverage will have the added benefit of encouraging an interdisciplinary mind set in participants, since they will gain first-hand experience with commonalities among the disciplines.

This approach is also consistent with our assumption that students from distinct scientific fields will take the core courses. Their diverse backgrounds will make it even more important that we not give preference to any particular discipline. Moreover, their presence offers the opportunity to form

interdiscplinary teams for course projects, which should further instill a broad view of science among participants. Goal-directed interactions with students from other fields should further benefit those who complete the science informatics curriculum.

3.5 Problem-Driven Content

We have claimed that there exist general principles of science informatics that cut across disciplines. These revolve around the types of structures and processes that arise in science, as well as the relations among them. Naturally, we hope to communicate these general principles to students in our courses, but we believe this is best done in the context of specific examples. One could easily do this in advanced, specialized courses, but we maintain that even core material should be organized around exercises and projects that give students hands-on experience.

Naturally, the organization of core courses around certain types of scientific activity will constrain the examples we select. For instance, a course on scientific data processing should expose students to data sets collected from particular scientific studies, whereas a course on scientific modeling should examine the details of specific scientific models that aim to account for such observations. Moreover, students should gain experience with specific informatics software that process data, interpret models, and support other key activities.

We recognize that many students from the sciences will not have the same programming skills as those with specialized training in computer science or informatics. As already noted, we do not think this will be a serious problem because there already exist many high-level software environments that address aspects of science that the curriculum should cover. Relying on these systems for exercises and projects should lower entry barriers and ensure the accessibility to a broad range of students, while still giving them direct experience with relevant scientific applications.

4. Structure of the Curriculum

Now that we have presented our views on science informatics and on desirable characteristics of courses in the area, we are ready to describe a specific curriculum. In this section we present the five core courses that we are developing, in each case discussing both the general aim and the content it would cover. We also discuss briefly other courses that students might take to complement them.

4.1 Introduction to Science Informatics

The sequence should begin with an introduction to science informatics that provides an overview of the entire field. This initial course should present science from a computational perspective, review the goals of scientific research, and examine the types of structures and processes that underlie the scientific method. Even students who do not complete the remainder of the sequence should come away with a basic understanding of science, the ways that informatics can aid its operation, and some experience with how specific techniques operate toward this end.

More specifically, participants who have taken this introductory course should understand the nature of scientific data, including computational methods for collecting, storing, manipulating,

analyzing, and visualizing it for scientific ends. They should appreciate the different forms of data that arise in alternative fields and their implications for processing. Equally important, students who have taken the course should understand the nature of scientific models and theories, including computational ways to represent them and use them for prediction or explanation. Again, they should recognize that disciplines specify models in different formalisms and utilize them in distinct ways, but that they always bear clear relations to the data they attempt to explain or predict.

Moreover, participants should become familiar with the computational nature of scientific method and practice. This includes the design and execution of experiments, analogous methods that arise in nonexperimental fields, and hypothesis testing and model evaluation in both contexts. They should also understand that scientific procedures can be quite complex and that scientific workflows can represent them formally and automate their execution.

Another important topic is the computational character of scientific discovery and innovation. Students who complete the course should understand the tasks of forming taxonomies, finding laws, constructing models, and inferring theoretical principles, as well as computational methods for approaching them. They should also become familiar with the communal context of scientific research, including the role of communication and interaction. This should include computational aids for retrieving and generating scientific papers, but also other ways to support interaction among researchers.

Most important, participants in the introductory course should understand the manner in which these facets of science interact and realize the potential for informatics tools to support the overall scientific enterprise. They should gain experience with specific informatics tools and their application in the physical, life, and social sciences in ways that demonstrate commonalities among thesm but that also respect their differences. In addition, students should understand how these informatics techniques relate to human scientists' activities and how they can help researchers achieve their goals more rapidly, accurately, and easily.

4.2 Scientific Data Processing

Courses that follow the introduction should delve into the major components of scientific research in greater detail. One such topic involves the character of scientific data and computational approaches to their representation, storage, retrieval, and processing. A preliminary Informatics course on storing and retrieving digital content would be useful background, but it should not be a formal prerequisite.

More specifically, participants who have taken this course should understand the variety of data types – both quantitative and qualitative – that arise across different scientific disciplines and how to encode them in digital form. They should also master basic techniques for storing and accessing such data, including design and use of databases, specialized file systems, and public repositories, along with the treatment of large-scale databases. Additional topics should include the knowledge-based annotation of scientific data (Jonquet et al., 2008), including the benefits and drawbacks of formal ontologies, their development, and their practical use.

Students should also learn about common practices for collection of data in both experimental and nonexperimental settings. This should include the design and use of measuring devices, including the role that models of observation play during data collection. They should also master the basic principles of experimental design and procedures for executing them. However, they should also learn that some fields cannot run controlled experiments and that one can still collect useful scientific data in purely observational contexts.

Another important topic concerns approaches to handling uncertain, missing, and potentially erroneous data. This should include basic methods for random sampling and repeated measures in experiments, as well as computational techniques for quantifying noise, tracking uncertainty, and imputing missing values (Schafer, 1999). Students should also learn about outliers and anomalies in scientific data, including techniques for detecting them (Breunig et al., 1999), and ways of handling them, and their role in driving the processes of theory revison and discovery (Darden, 1991).

More advanced issues involve the problems associated with scientific data that are heterogenous in form and content. In this area, participants should understand the need to combine observations that occur at different temporal and spatial scales, that arise at separate organizational levels, and that are based on distinct observational assumptions. Naturally, they should also learn about computational methods that address these challenges (Gross & DeAngelis, 2001).

In addition, scientists often attempt to visualize their data in various fashions, and informatics offers techniques that support this process in flexible ways (Telea, 2008). Students should learn about these approaches to visualization, including their role in classifying observations, detecting anomalies and outliers, recognizing relationships among variables and other patterns, and analyzing scientific data more generally.

Of course, participants in this course should get experience with informatics tools for data storage, preparation, and analysis in the context of representative problems from the sciences. These should include examples from a number of distinct fields that involve data with different characteristics and that raise different challenges in representation and processing. For example, fields like astronomy have access to large data sets, which raises separate issues from ones like biology in which data are often rare and expensive to collect.

4.3 Scientific Modeling

Although science relies centrally on data, it depends equally on models and theories that account for those data. Thus, we propose another course on scientific models, including their representation, their use in simulation and explanation, and their evaluation in relation to observed data and phenomena. Naturally, it should also examine informatics systems that automate or assist in model storage, construction, use, and evaluation.

Participants who have taken this course should understand the role of models in science, especially their ability to interpret and account for observations. They should also recognize the distinction between descriptive laws, which merely summarize or predict data, and theory-based models, which explain observed phenomena in deeper terms. In addition, they should be familiar with the diversity of models that arise in different scientific disciplines and ways to represent them formally in computational terms.

In particular, students should understand both structural models, which emphasize static relationships among a set of components, and mechanistic models, which emphasize change in entities over time. Both types of models can take either qualitative or quantitative form, with the latter being more precise but also more difficult to specify. Moreover, participants should learn about the causal character of models, their ability to incorporate unobserved or theoretical terms, and the distinction between exogenous and endogenous variables.

In addition, those who complete the course should master computational methods for using scientific models to account for known phenomena. These should include techniques for interpreting structural models, both qualitative and quantitative, to explain observations or to make predictions about them. They should also include computational methods for interpreting and simulating causal and mechanistic models, whether they take qualitative or quantitative form.

One important special case involves computational techniques for simulating a special class of causal models that are cast as sets of differential equations, which one can use to predict how variables change over time. However, students should also learn how one can link such equations to notions of process and mechanism that provide deeper accounts of dynamic systems (Bridewell et al., 2008). They should also recognize that differential equations are only one type of scientific model, and that other formalisms are more appropriate for some disciplines.

Another important topic involves the evaluation of scientific models. Traditional treatments assume this should focus mainly on their ability to predict or explain observations, but other factors include a model's comprehensibility (Langley et al., 2002), simplicity (Simon, 1968), and consistency with existing knowledge (Pazzani et al., 2001). The latter also touches on how models relate to more general theoretical principles and constraints that hold across an entire discipline.

Again, participants should garner experience with informatics tools for creating, storing, and using models on relevant problems from a variety of scientific fields. These examples should reflect the diversity of model forms and the ways that researchers use them. For example, some fields rely primarily on quantiative models that support precise predictions of change over time, while other disciplines typically assume qualitative models that explain phenomena but are weaker in predictive abilities, but both are legitimate and useful forms of scientific knowledge.

4.4 Scientific Discovery and Creativity

Traditional treatments of science emphasize the role of laws and theories in predicting or explaining observations, as well as the use of data to evaluate candidate laws and theories. However, they downplay an equally important part of the scientific enterprise: the creation and discovery of new lawful and theoretical knowledge. This activity is broad enough in its own right to deserve a separate course in a curriculum on science informatics.

Participants who have taken this course should understand the contribution of creativity and discovery to science, as well as its inherently computational character. They should also appreciate the roles in the discovery process of heuristic search through a space of laws, hypotheses, or models and retrieval of structures from memory. They should also understand how these mechanisms are closely related to similar processes that underlie problem solving in everyday settings.

In particular, students should understand computational methods for discovering descriptive scientific knowledge. These should include techniques for constructing taxonomies and ontologies from observed entities, which generally occurs early in a field's development. They should also master mechanisms for creating descriptive laws which specify qualitative relations that summarize observations (Lee et al., 1998). Finally, they should become familiar with methods for another class of descriptive regularities – numeric laws – that summarize data in greater detail (Langley & Żytkow, 1989).

In addition, course participants should master information-processing approaches to creating explanatory scientific knowledge that accounts for observations at a deeper level. These should include methods for model construction in science, such as techniques for creating and revising explanatory accounts from background knowledge and data (Asgharbeygi et al., 2006). An important related topic involves techniques for inferring theoretical concepts, processes, and principles that appear in such explanatory models.

The course should also cover computational approaches to designing scientific measuring instruments, including the ways they take advantage of existing measuring devices, laws, and models. Moreover, it should introduce the continuum between normal and revolutionary science (Kuhn, 1970), and it should examine information-processing accounts of their differences. Finally, it should examine how these varied aspects of discovery interact, including ways to integrate them into computational systems that support a wide range of scientific activities.

As in other courses, participants should acquire expertise with informatics systems that automate, assist, or model the creation of knowledge, along with their application to important problems in the natural, life, and social sciences. These examples should cover a wide range of discovery types and they ways they arise in different scientific fields. For example, some disciplines focus on knowledge-guided construction of explanatory models, whereas other fields emphasize data-driven induction of descriptive laws, but both have legitimate roles to play in scientific discovery.

4.5 Scientific Communication and Interaction

Standard analyses of science focus on the individual researcher, and many tools for science informatics reflect this bias. However, the scientific enterprise is inherently communal, and it is crucial that future researchers appreciate this fact and become familiar with technologies that support this property (Schneiderman, 2008). Thus, our curriculum will also include a core course on scientific communication and interaction.

Participants who have taken this course should understand the social character of science, including the types of entities that comprise it, the relations that can link those entities, and the activities that they pursue during interactions. They should also master concepts related to social networks well enough to analyze specific scientific communities and suggest ways to promote productive, collaborative, and interdisciplinary scientific research.

Perhaps the most visible form of scientific communication occurs through refereed publications in journals and proceedings. Thus, students should learn about publication practices within scientific communities, how they implement such practices, and how they share published and even unpub-

lished works among colleagues. Moreover, they should master informatics technologies that support the distribution and access of documents, along with methods for retrieving content from both unstructured and structured sources. These should include electronic journals, citation databases (e.g., PsycINFO and ISI Web of Knowledge), literature archives (Giles et al., 1998), and other material available on the World Wide Web.

We can differentiate between two broad forms of interaction among scientists – cooperation and competition – often involving the same entities. Both can occur at the level of individuals, projects, laboratories, or larger groups, and each has major implications for scientific practice. Scientists often want to share data and knowledge, including their observations, models, predictions, explanations, and meta-level content, and informatics technology can support these intentions, but researchers also want to retain certain information to retain an edge over others (Pearson, 2003).

Participants in the course should learn about such cooperative and competitive behavior in science, as well as informatics tools that operate within their constraints. They should become familiar with privacy concerns for the producers and consumers of scientific information and ways to safeguard this content as necessary (Bertino & Sandhu, 2005). In particular, they should master techniques for supporting information security as it relates to data availability, to user-level access, and to record anonymizing (LeFevre et al., 2005).

Finally, students should learn about the broader role of science within society and how it helps inform personal and public decision making. This should include types of scientific results that can influence personal choices and public policy, as well as informatics techniques that make such content available and help select among alternatives. In general, they should learn about relationships among scientific researchers, policy makers, and stakeholders, and they should become familiar with computational methods that facilitate communication and other interactions among them.

4.6 Additional Courses

To complement the core courses we have just described, we believe it is important that science informatics students also take additional courses that provide them with broader context on the one hand and more detailed content on the other. The former goal seems best achieved by requiring a course on the philosophy of science, which typically takes a prescriptive or normative approach to research, and a second on the history of science, which takes a descriptive approach that attempts to characterize how scientific research actually operates. Both should be generic courses on these topics that include examples from many disciplines, which should reinforce the interdisciplinary perspective offered by the core curriculum.

However, our goal is not to train generalists who understand the abstract principles of science informatics in the absence of domain content. The apprentice scientists who complete the curriculum will go on to careers in industry or academia that require them to use the computational skills they acquire in some established scientific field. Thus, we assume they will obtain a degree in this discipline that will educate them in the relevant material and provide the background knowledge needed for effective use of computational methods.

We should also encourage these students to take advanced courses in the application of informatics to their chosen area, and many such offerings are already available at Arizona State University and

other campuses. Nevertheless, we maintain that it is important that they take these domain-specific informatics courses after they have completed at least some of the core classes. This will help them understand the place of domain-specific techniques within the broader context of science, identify ways to adapt methods from other disciplines to their own, and instill appreciation and respect for other fields that is often lacking in scientists who specialize too early.

5. Curriculum Outcomes and Assessment

The curriculum in science informatics that we have described aims to prepare students for future careers as scientists who can take advantage of existing informatics systems, adapt readily to new ones as they are introduced, and even contribute to the development of new information technologies to support scientic research. In this section we discuss the learning outcomes that should result from the curriculum, along with our plans for assessing them.

5.1 Expected Outcomes

We expect that students who complete the above courses will acquire essential knowledge and skills related to science informatics. For example, they should achieve a broad understanding of the scientific method, including the structures and processes that comprise it and how they relate to each other. They should also be able to think about and describe these elements and activities in computational terms, which they can in turn use to characterize informatics technologies that automate or support them.

In addition, students who complete the curriculum should become familiar with specific tools for science informatics, learn to identify which ones are relevant to given aspects of the scientific enterprise, and become adept at using them to achieve particular research objectives. They should gain the ability to master new informatics tools rapidly, identify their strengths and weaknesses, and propose changes to their design that would improve them. Students should also understand how different tools complement each other and how to combine them creatively to achieve results that they could not manage alone.

Moreover, students should understand how techniques from science informatics address the same tasks as human researchers, explain the ways in which their operation is similar to and different from human cognition, and identify facets of science that benefit from computational automation or assistance. Finally, they should gain knowledge about and respect for a variety of scientific disciplines based on their experience with informatics problems in fields of research that are superficially different but that have underlying commonalities.

5.2 Plans for Assessment

A successful curriculum also depends on a well-defined plan assessing student outcomes. To determine whether students can identify and characterize the computational aspects of science, we will include questions that test this ability in the final examinations for each course. In addition, we will administer a questionnaire for self assessment after students complete the sequence of core courses. Questions will address their appreciation for broad themes in science and their confidence in using

informatics tools. We will also interview students individually to assess their grasp of informatics ideas with respect to their major fields of study.

Furthermore, we intend to measure each student's ability to identify informatics tools relevant to given scientific problems and to adapt them as necessary. To this end, the core courses will incorporate case studies that require selection from a set of available systems and their utilization to answer particular scientific questions, with student evaluation based on appropriateness of their selection and their manner of use. Each course will also include an exercise that requires students to address a scientific problem in one field by adapting an informatics system originally designed for another discipline, which we will evaluate in a similar manner.

Finally, we will require students to maintain portfolios of their coursework that we will examine after they complete their second and final core courses. We will analyze these portfolios for evidence of broad understanding and appreciation of structures and processes that arise in science, the computational character of scientific research, the operation of specific informatics tools and their relation to human scientific behavior, and the common goals and methods of different disciplines. We will use results from this analysis to identify changes to the content and organization of particular courses that would improve their effectiveness.

6. Concluding Remarks

In the preceding pages, we have argued that scientific practice is becoming increasingly reliant on information technology, and that there is a growing need for scientists who are trained in its principles and operation. We defined *science informatics* as the use of computational metaphors and methods to understand and support scientific research, and we presented a scenario for one possible future this technology could support. However, this vision depends on having a cadre of scientists who can use the technology of science informatics effectively, which means that we must educate the next generation in its use.

We also made a number of claims about this emerging field. In particular, we argued that informatics tools can support every facet of the scientific enterprise and, more radically, that science has always been a computational endeavor, with recent software improving its operation but not changing its basic nature. Moreover, we claimed that this topic is best approached by identifying the component tasks in science and examining the types of informatics systems that can contribute in each case. Finally, we argued that most principles of science informatics, and even many specific computational techniques, are general enough to contribute across different research disciplines.

Next we considered some characteristics of a curriculum on science informatics suggested by these claims that would educate scientists in this area. We stated that courses should be at an intermediate level that both upper-division undergraduates and junior graduate students can take. We also argued that the content should be accessible to students with diverse backgrounds, relying on minimal prior training in computing and informatics. The curriculum should be organized not around specific information technologies or by scientific disciplines, but around component tasks that arise across different fields. Courses should present general principles but illustrate them by providing hands-on experience with specific tools on particular problems, but these examples should come from multiple scientific fields to instill an interdisciplinary attitude in students.

In this context, we proposed a curriculum in science informatics that reflects these characteristics. Students would start with an introductory course on the topic that provides an overview of the scientific enterprise from a computational perspective. After this, they would take courses on the storage, retrieval, and analysis of scientific data, on the representation of scientific models and their use for prediction and explanation, on the discovery of scientific taxonomies, laws, models, and theories, and on communication and interaction within scientific communities. In addition, the curriculum would include general courses in the history and philosophy of science, followed by optional advanced courses on informatics applications to particular fields. We also outlined the outcomes we expect from the curriculum and our plans for assessing its results.

We are developing these courses under the auspices of a broader informatics program at Arizona State University, which provides strong encouragement for interdisciplinary, problem-driven education and research. But this program, however successful, cannot meet the increasing demand for scientists who are trained in the effective use of information technology. Such researchers must become the default rather than the exception if science is to take advantage of informatics breakthroughs, and we need similar programs at many campuses to educate them. We hope that other universities will adopt and implement our proposal for a curriculum on science informatics, and thus address this growing challenge.

We believe that such a concerted effort will transform science from its current state, where most researchers rely on software experts for their computational needs, to one in which the majority are adept consumers of informatics technology that boosts their effectiveness many fold. Educating the next generation of researchers in science informatics will revolutionize the scientific enterprise, not by changing its basic structures and processes, but by enabling its members to use computational aids that let them operate far more efficiently and accurately than in previous eras.

Acknowledgements

This work was supported by a gift from Microsoft Research and by NSF Grant No. IIS-0326059. We thank Sethuraman Panchanathan for proposing a degree program in science informatics, as well as Yan Xu and Tom McMail for discussions that helped refine the ideas presented in this paper.

References

Altschul, S. F., Boguski, M. S., Gish, W., & Wootton, J. C. (1994). Issues in searching molecular sequence databases. *Nature Genetics*, 6, 119–129.

Asgharbeygi, N., Bay, S., Langley, P., & Arrigo, K. (2006). Inductive revision of quantitative process models. *Ecological Modelling*, 194, 70–79.

Bertino, E., & Sandhu, R. (2005). Database security — Concepts, approaches, and challenges. *IEEE Transactions on Dependable and Secure Computing*, 2, 2–19.

Breunig, M., Kriegel, H. P., Ng, R. T., & Sander, J. (1999). OPTICS-OF: Identifying local outliers. *Proceedings of the Third European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 262–270). Berlin: Springer–Verlag.

Bridewell, W., Langley, P., Todorovski, L., & Džeroski, S. (2008). Inductive process modeling. *Machine Learning*, 71, 1–32.

Cohen, S., & Hindmarsh, A. (1996). CVODE: A stiff/nonstiff ODE solver in C. *Computers in Physics*, 10, 138–143.

Darden, L. (1991). *Theory change in science: Strategies from mendelian genetics*. New York City, NY: Oxford University Press.

Giles, C. L., Bollacker, K. D., & Lawrence, S. (1998). CiteSeer: An automatic citation indexing system. *Proceedings of the Third ACM Conference on Digital Libraries* (pp. 89–98). Pittsburgh, PA: ACM Press.

Gross, L. J., & DeAngelis, D. L. (2001). Multimodeling: New approaches for linking ecological models. In J. M. Scott, P. J. Heglund, M. Morrison, M. Raphael, J. Haufler, B. Wall (Eds.), *Predicting Species Occurrences: Issues of Scale and Accuracy*. Covello, CA: Island Press.

Jonquet, C., Musen, M. A., & Shah, N. H. (2008). A system for ontology-based annotation of biomedical data. *Proceedings of the International Workshop on Data Integration in the Life Sciences* (pp. 144–152). Evry, France: Springer–Verlag.

Kuhn, T. S. (1970). *The structure of scientic revolutions*. Chicago, IL: University of Chicago Press.

Kulkarni, D., & Simon, H. A. (1988). The processes of scientific discovery: The strategy of experimentation. *Cognitive Science*, 12, 139–175.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). *Scientific discovery: Computational explorations of the creative processes*. Cambridge, MA: MIT Press.

Langley, P., & Żytkow, J. M. (1989). Data-driven approaches to empirical discovery. *Artificial Intelligence*, 40, 283–312.

Langley, P., Shrager, J., & Saito, K. (2002). Computational discovery of communicable scientific knowledge. In L. Magnani, N. J. Nersessian, C. Pizzi (Eds.), *Logical and computational aspects of model-based reasoning*. Dordrecht: Kluwer Academic Publishers.

Lee, Y., Buchanan, B. G., & Aronis, J. M. (1998). Knowledge-based learning in exploratory science: Learning rules to predict rodent carcinogenicity. *Machine Learning*, 30, 217–240.

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2005). Incognito: Efficient full-domain K-anonymity. *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data* (pp. 49–60). Baltimore, MD: ACM Press.

Pazzani, M. J., Mani, S., & Shankle, W. R. (2001). Acceptance by medical experts of rules generated by machine learning. *Methods of Information in Medicine*, 40, 380–385.

Pearson, H. (2003). Competition in biology: It's a scoop! *Nature*, 426, 222–223.

Presidents Information Technology Advisory Committee (2005). *Computational science: Ensuring americas competitiveness*. National Coordination Office for Networking and Information Technology Research and Development, Arlington, VA.

Schafer, J. L. (1999). Multiple imputation: A primer. *Statistical Methods in Medical Research*, 8, 3–15.

Schneiderman, B. (2008). Science 2.0. *Science*, 7, 1349–1350.

Simon, H. A. (1968). On judging the plausibility of theories. In B. van Rootselaar and J. F. Stall (Eds.), *Logic, methodology, and philosophy of science III*. Amsterdam: North-Holland Publishing.

Telea, A. C. (2008). *Data visualization: Principles and practice*. Wellesley, MA: A. K. Peters, Ltd.