Using provenance to manage knowledge of In Silico experiments

Robert Stevens, Jun Zhao and Carole Goble

Submitted: 15th January 2007; received: 18th March 2007

Abstract
This article offers a briefing in one of the knowledge management issues of in silico experimentation in bioinformatics. Recording of the provenance of an experiment—what was done; where, how and why, etc. is an important aspect of scientific best practice that should be extended to in silico experimentation. We will do this in the context of eScience which has been part of the move of bioinformatics towards an industrial setting. Despite the computational nature of bioinformatics, these analyses are scientific and thus necessitate their own versions of typical scientific rigour. Just as recording who, what, why, when, where and how of an experiment is central to the scientific process in laboratory science, so it should be in silico science. The generation and recording of these aspects, or provenance, of an experiment are necessary knowledge management goals if we are to introduce scientific rigour into routine bioinformatics. In Silico experimental protocols should themselves be a form of managing the knowledge of how to perform bioinformatics analyses. Several systems now exist that offer support for the generation and collection of provenance information about how a particular in silico experiment was run, what results were generated, how they were generated, etc. In reviewing provenance support, we will review one of the important knowledge management issues in bioinformatics.

Keywords: In Silico experiments; provenance; workflow; data derivation; validation and verification of results

INTRODUCTION
This article provides a briefing on the provenance of in silico experimentation—a largely unexplored area of knowledge management in bioinformatics. Biology can be said to be a knowledge based discipline [1, 2] and consequently bioinformatics can be said to be a knowledge management discipline. That is, everything in bioinformatics, particularly at the informatics end, is close to knowledge management.

A considerable amount of effort in bioinformatics is expended in both storage and analysis of data and its associated knowledge. It is storage, however, where most effort in management of the discipline’s knowledge has taken place. Although many bioinformaticians spend a great deal of time analysing data, virtually no effort is expended in managing the knowledge about these in silico experiments themselves. It is this neglected area of knowledge management in bioinformatics that this article addresses.

The recording of the details of an in silico experiment has become known as provenance [3, 4]. The provenance of something is its origin and history;
how did this thing come to be here as it is now?

Provenance is usually a term used in art and antiques, but it is easy to see how we should be interested in the origin and history of the data we use in bioinformatics. Both in literature and our databases, we as scientists are keen to know whence facts and data came; it is an important part of the scientific process.

The recording of provenance, what was done, where, when, how and why in a scientific experiment is given in any science laboratory. The idea of performing a long and costly experiment at the bench and not recording all aspects of that experiment is unthinkable to most scientists. These recordings or provenance, are vital for validation and verification of an experiment, both by the scientist and the wider community. The recording of provenance for an experiment provides scientists with context for assessing that experiment and consequently confidence in the scientific process [5].

This has not been the case for in silico experiments. Historically, single sequence orientated bioinformatics was performed by cutting and pasting between web-based tools [6]. In such a situation, there is little perceived need for recording. It is easier to re-create than it is to record. Over time, however, bioinformatics experiments have become both more complex and larger. Cutting and pasting is neither sustainable; nor is it easy to re-do an analysis without a record of what has happened.

As well as scientific best-practice, recording is necessary due to the nature of bioinformatics analyses. Bioinformatics data are such a mess that much of a bioinformatician’s time is spent husbanding knowledge: mapping identifiers from one resource to another; transforming data between proprietary formats, etc. All of these actions need to be recorded if experiments are to be validated and verified.

Post-genomic bioinformatics; high-throughput experimentation and eScience have all come together to mean large-scale in silico experimentation is both necessary and now possible. The nature of bioinformatics is changing and with that change the need for adopting scientific best practice for in silico experiments will become more pressing. We will visit these issues throughout this article. In ‘Bioinformatics in silico experimentation’ section, we further describe the motivation for collecting provenance and the nature of bioinformatics experiments. After discussing the tasks enabled by provenance information, in ‘Models of provenance’ section we talk about models of provenance information that can be used to describe and manage the provenance described in ‘Bioinformatics in silico experimentation’ section. In ‘Systems implementing provenance’ section, we briefly introduce some provenance systems and discuss provenance as a knowledge management issue in ‘Discussion’ section.

**BIOINFORMATICS IN SILICO EXPERIMENTATION**

Bioinformatics analyses or experiments are typically data flow pipelines [7]. Data are taken from one resource (be it a user or database) and delivered either to another tool or used to retrieve related data from another data resource. So, a bioinformatics experiment is typically some data passed between a series of data resources and tools. An analysis can be as little as one tool, but are often much more complex [8].

Actions within bioinformatics experiments broadly fall into three categories [8]: Data are filtered, such as retrieving a particular database entry. Data are transformed, such as turning a set of sequences into a multiple alignment or a multiple alignment into a phylogenetic tree. Finally, data can be both filtered and transformed, such as filtering a sequence database for similar sequences and transforming into a collection of similarity records [8]. Any pipeline will consist of a series of such stages.

As data are passed between resources there is often amplification of those data. One sequence becomes many tens of similarity entries. One query to a database can become scores of entries. Projections over any of these data will transform one entity (a database record) into many parts of that entry (species, sequence features, etc. for example). This amplification will mean that iterations will often happen within experiments. As data are filtered, transformed and amplified it is important that the data derivation path is maintained. This is particularly important when the identifier of an entity in one resource is mapped to the same entity’s identifier in another resource—an all too common issue in bioinformatics.

Many autonomous groups create the wealth of bioinformatics resources [9]. One consequence of this autonomy is that bioinformatics resources are highly heterogeneous [10, 11]. There is,
for instance, the Seqret programme (http://emboss.sourceforge.net/apps/seqret.html) that converts between the 28 different formats for biological sequences such as nucleic acids and proteins. Each database has its own proprietary ASCII format or schema. This autonomy and heterogeneity mean that there is no common typing mechanism within bioinformatics.

**Implementing In Silico experiments**

There are three ways in which the data pipelines that are most in silico experiments can be performed:

**Human driven**

A human uses one or more web pages to perform an experiment. Data are cut-and-paste between files and pages.

**Bespoke programmes**

Most programming languages offer facilities that enable the inclusion of a variety of external resources. It is possible to call command-line programmes; web pages; databases—often in a distributed setting. This is made easier by the programmatic access to bioinformatics resources that are becoming more common [12]. Many resources offer bespoke API, but web services (http://www.w3.org/TR/wsdl20/) implementations are becoming popular. These offer the possibility of a Service Orientated Architecture (SOA) [13].

**Workflow**

Workflow systems such as myGrid, Kepler, ISI and Chimera are becoming popular across science as people endeavour to develop eScience technologies [6, 14–16]. Workflows are explicit and exchangeable scripts for interoperating and linking these services [17]. Ideally, in a workflow system, a user simply joins services together to form their experiment. This higher level view of the process makes it easier for scientists to describe and run their in silico experiments in a structured, repeatable and verifiable way. Workflows are usually assembled and managed from a workbench and execution environment. Typically, a workflow comes with a visual workflow representation, such as seen in Figure 1, which helps to understand its behaviour and explain it to others. myGrid (http://www.mygrid.org.uk) is a project whose aim is to provide a set of services to support in silico experiments in bioinformatics. myGrid’s flagship application is Taverna [7] which allows biologists to create and run workflows

---

**Figure 1:** A workflow for finding and analysing extensions to a genetic map for Williams Beuren syndrome from the myGrid project [6]. In this workflow, a DNA sequence is masked for repeats and submitted to a BLAST search. Subsequently, genes are predicted in regions that are found to extended the query sequence and proteins predicted in these genes are further characterised in a further workflow [6].
over some 3500 bioinformatics services [3]. m\textsuperscript{y}Grid workflows have been used in completing genetic maps, analysing the genetic basis of diseases, docking of small molecules in drug design studies and a variety of systems biology experiments [3]. One of the major services provided by m\textsuperscript{y}Grid to support \textit{in silico} experiments as workflows is the provenance service [3, 6, 18]. We will use m\textsuperscript{y}Grid’s approach to provenance to illustrate this briefing, but draw in other systems’ approach to provenance as appropriate.

**Knowledge of experiments**

Ideally, a bioinformatician would wish to make recordings about the materials and methods used in any of the ways of performing an \textit{in silico} experiment. For human driven experiments the reliance falls upon the human to note in a log-book or save files, etc. Human nature being what it is, this usually does not happen. So each of the experimental features described earlier should be recorded as part of the materials and methods—part of the provenance of an experiment or analysis. It will be important for a user to be able to find out what has happened to his or her data at any one point within the flow of an analysis. Knowing that one protein sequence, for instance, was passed to a BLAST service with a particular set of parameters and produced an input database filtered on sequence similarity, against a certain taxa with similarity settings \(x, y\) and \(z\) should be important for the verification and validation that is part of scientific best practice.

Such provenance recording can be a component of any automated \textit{in silico} experiment. A provenance service could be part of an application built upon services, as well as in a workflow system such as m\textsuperscript{y}Grid [3, 7]. The means to record provenance can be built into both applications and workflow systems. In the latter, however, the inherent flexibility of a system that allows arbitrary experiments to be designed and run means that a consistent (within that experiment system) model of provenance is used.

Experimental procedures are implicit for application style or bespoke programmes that instantiate and automate particular experimental protocols. What is done to perform an analysis is hidden within that programme. It is possible for it to be made manifesto the user, but this is not the usual case.

A significant aspect of workflows is that they make explicit the process employed in analysing data. Instead of being hidden behind a user interface, the workflow is the user interface (see Figure 1). It is not only the order of bioinformatics services, but also the helper services used to couple the heterogeneous resources together [19]. It is thus possible to inspect exactly what techniques are employed to elucidate biological facts, but also what transformation and mappings have occurred in order to achieve the analysis.

With workflow systems, all of this is made explicit or transparent in that a user can see through to what is actually happening to the data. These explicit descriptions of work can be exchanged and discussed in their own right [17]. A standard script or application might achieve the same analytical goal, but the means is hidden.

Thus workflow descriptions of \textit{in silico} experiments are themselves a form of knowledge capture in their own right. There is a move to develop workflow repositories in which descriptions of experiments can be described and stored for users to later find and use [20]. Such repositories are a form of knowledge management for \textit{in silico} scientific protocols. The understanding of how to perform an analysis or how a particular analysis was performed is important and useful knowledge. For human driven analysis, using web pages, knowledge of the experimental procedure lies within the user’s head, in some text or, if luck holds, written in a log-book. It is not captured according to any standard model or schema and relies on good practice. Nor is it available for analysis, querying, exchange, etc.

As experiments are performed or run via a computer, what is done during an experiment should be logged. This should be easier than for a wet laboratory experiment. In such experiments, there is inherently much more reliance upon the human part of the experiment (unless it is on a single machine). Given that \textit{in silico} experiments are necessarily run on a computer, it should be possible to record a great deal of what is needed in provenance automatically. The informatics question is what should be recorded and what model should be used for provenance records?

**Knowledge around experiments**

The experiment itself is the core of provenance recording. It is the method part of the materials and methods for an \textit{in silico} experiment. Naturally, we wish to capture other aspects of the experiment,
such as why the experiment was performed (hypothesis or goal); which resources were used; what happened during the experiment; how data were related to each other who created and ran the experiment; what were the observations, conclusions and findings, etc. All of these aspects are knowledge around the experiment; they are its context. This contextual knowledge is also provenance.

In Silico experiments can be large, complex and run many times, both with the same data and different data inputs. These provenance data should be recorded consistently between experiments and allow aggregation across experiments. For example, the same data and services will appear in many different experiments. Results or intermediate data from one experiment will be used as inputs to another. Experiments will be re-run from intermediate points after failure or a change in circumstances. In analysing data, for the provenance of an experiment becomes data in its own right, aggregations of identical data holdings will facilitate recognition of patterns and facilitate querying. This requirement relies on consistent, unique identity of entities within data holdings.

Once this knowledge about an experiment or its provenance has been gathered and stored, it must be able to be used by scientists; that is, queried and re-used as inputs to further experiments [5]. The provenance of an experiment is a form of knowledge about that experiment and scientists could explore provenance knowledge to support the following kinds of tasks [3, 18]:

- Debugging experiments from a log of events.
- Recording what services were accessed and with which data.
- Comparison between different protocols on the same data or the same protocol on different data.
- Validity checking of novel results to ensure it is worthy of further investigation before they commit to expensive laboratory-based experiments based on these results.
- Tracking the implications of updates when a service or data set used in the production of a result changes.
- Supervisors and laboratory heads browse it to summarise progress and to aggregate across it from all their researchers.
- Service providers aggregate process-centric provenance information together intelligence about their services performance and patterns of use.

- Third party research groups and regulatory authorities, who need to trust the validity of results, want a detailed, accurate and reproducible audit of the experiment and data outcomes.

Questions such as these could exploit the provenance of analyses in order to discover knowledge about the experimental context in which scientists work.

MODELS OF PROVENANCE

We have established a motivation for recording provenance and talked generally about the experimental context that provenance has to represent. In this section we look in more detail about what kinds of information should be recorded and how it can be modelled.

Different kinds of provenance

There are different kinds of provenance that can be collected. Figure 2 shows myGrid’s view of the different kinds of provenance recording made during an experiment [18].

Figure 2: The provenance pyramid. A bottom up arrow from the process view to the data view shows the data provenance is derived from the process provenance. A bi-directional arrow from the organisational provenance to the process and data provenance shows that the organisational metadata can be attached to anything in the process and data provenance information. A bottom-up arrow from the organisation and data view, to the knowledge provenance, shows the knowledge provenance is the top level view of the primary provenance (i.e. the process, data and organisational provenance), which contains users’ abstraction and interpretations.
Process provenance
Process provenance is similar to traditional event logs, records in which each experiment performed, the order of services invoked and data processed or produced by each service invocation is recorded with the time-stamps of service invocations and the workflow run.

Data provenance
Data provenance is about the origin of a data product, either a workflow input or output or an intermediate data product. It tracks the data derivation path of a data product, inferred from the process level provenance.

Organisation provenance
Organisation provenance states the creator of a data product, a service, a work flow; the launcher of a run; the project an experiment belongs to; or the hypothesis this experiment or project is based upon, etc.

Knowledge provenance
Knowledge provenance provides either a personal, more abstract view over the logging and derivations, or a domain-specific understanding about the data processed or generated during the runs. The first type is experiment-specific and person-specific and contributed by the scientists. The second type is either contributed by scientists or data curators annotations about the data.

Another case is the discrimination between external and internal provenance [3, 5]. Bioinformatics analyses are inherently distributed. There will be provenance that comes from these external resources. The workflow in Figure 1 uses the Genbank database to retrieve a DNA sequence. The provenance fragment for this workflow shown in Figure 3 shows the GenBank sequence urn:genbank:50 that is extracted from the GenBank sequence database is annotated with the external knowledge that, this sequence is extracted from the organism ‘Homo sapiens,’ published in Nature 2003, has a pubmed reference number 12853948, and the latest version is version 6. Other provenance is internal to the system governing the experiment, such as creator of the experiment, derivation paths, etc. Figure 3 shows a fragment of the provenance collected for the workflow.

Figure 3: A fragment of the provenance graph for the workflow shown in Figure 1. In this example, process level provenance shows the BLAST service taking data2 (urn: data:2) and outputs data3 (urn: data:3). The data level provenance shows the derivation of the pair wise alignments in data collection1 (urn: dataCollection:1) from data3, the output of process BLASTInvocation3 (urn: service: BlastNInvocation:3). We can also see organisational level provenance like the experiment’s creator (urn: person:1). Finally, the user can supply knowledge level provenance, such that data 2 is a DNA sequence.
in Figure 1 and we can see that the collection of alignments (data collection1) is derived from data 3.

Collecting provenance
We have described higher-level views of what can be provenance and what should be collected in terms of kinds of information. Now we move from high-level descriptions to lower levels of provenance data.

The Provenance Aware Service Oriented Architecture (PASOA) project (http://www.pasoa.org/) is part of the EU Provenance project (http://www.gridprovenance.org/). The goal of PASOA is to build a provenance infrastructure for recording and preserving provenance using an open provenance protocol and to provide some information about the quality and accuracy of results and services by some reasoning process [21]. The provenance model of PASOA, the p-assertion schema, is process-oriented and contains three parts: interaction p-assertions, service state p-assertions and relation p-assertions [21]. The first is internal provenance, being information about how actors such as services interact via inputs and outputs. Interaction P-assertions of this type are inferred by the system driving the experiment. The last two are external provenance, showing first the performance of the process yielding data (such as timings, etc.) and the second how the service obtained the output data by applying some functions to the input data, such as mapping codons of a nucleic acid sequence to an amino acid code.

It can be seen that these assertions fall mainly into the process and data types of provenance described earlier. The collecting of these assertions about how an in silico experiment was performed provides the basis of provenance.

Building a web of provenance
These statements about the experiment are easily presented as triples. 'Process y has input x'; 'Process x has output z'; 'datum a was derived from datum b', etc. Such triples naturally form a graph [5]. Projects such as myGrid have used the Resource Description Framework (RDF) (http://www.w3.org/RDF/) to represent their provenance in this way.

This graph can also be regarded as a web of science, as proposed by Hendler [22]. The description of an experiment and all its information, integrated with other experiments, form a large graph or web, just as the Web itself is a large graph. The provenance fragment shown in Figure 3, when repeated many times for the same and other experiments, will form such a web of science. This graph can be explored and queried by scientists in support of the experiment life-cycle. To truly form a 'web of science', the process and data level provenance should be combined with organisation and knowledge level provenance to give a rich view of an experiment's context.

The heterogeneous nature of bioinformatics data described in ‘Bioinformatics in silico experimentation’ section means that data need to be ‘massaged’ when transferred from one service to another. These shims map between different identifiers; transform formats; and decompose complex data [19]. The provenance graph needs to be maintained across the actions of such shim services and over the amplifications of data mentioned in ‘Bioinformatics in silico experimentation’ section. The data derivation needs to be maintained during these operations, particularly in the last case where complex data are decomposed, as in extracting a protein sequence from a protein database entry. Similarly, as one data input can produce many data outputs, the derivations must be maintained. These are vital for the verification and validation of results.

Finally, the organisation and knowledge level provenance suggested for myGrid needs to be supplied. Knowledge level provenance is especially relevant in bioinformatics, where there is no widely used, common data typing and many service’s parameters are simply typed as string despite representing, for instance, integers or complex data types such as databank entries. In addition, the task performed by a service, the contents of data resources and the relationships between inputs and outputs are not described. Ontologies are proposed as a solution for this kind of knowledge provision [23]. The myGrid project uses a service ontology to provide this support [24]. Both knowledge and organisation provenance are mostly provided by the user through the host systems that enacts the experiment. The descriptions of what significant nodes and arcs represent in terms of the biology and bioinformatics also potentially shield the scientist from having to interact with low-level process and data assertions that might be overwhelming in many cases. They would, however, remain to be used as and when necessary.

The identity crisis in bioinformatics
The assembly of such a graph or web depends on identifying entities within that web tools, data
resources and data items need unique, stable identifiers. Creating a coherent provenance graph or ‘web of science’ needs aggregation. We need, for instance, to know when:

- Two experiment protocols are the same.
- Input data items supplied by a user are the same.
- Processes used by a variety of experiments are identical.
- Data outputs from a process are identical.

In the provenance fragment of Figure 3, each node will have its own identifier. Imagine that the same workflow is run many times with different data input. The same BLAST service is used each time and so has the same identity. The different data inputs will have different identifiers. Consequently, there should be aggregation of all the separate provenance recordings of these experiments about the BLAST service.

One Uniprot record, for example, might be used as an input or output in many experiments. The same sequence might be supplied to a BLAST service and produce the same set of alignments each time; it might also produce different alignments as databases are updated or other parameters are changed. Consequently a reliable identification mechanism is needed to make provenance data work to its best extent. It is difficult to ask a question such as ‘find all occurrences of datum x’ when that datum might appear with different identifiers.

Managing identity for bioinformatics provenance proves something of a problem [25]. It is difficult to know without an unjustifiable amount of computation and guesswork when two database records, for instance, represent the same entry. The same goes for the contents of results collections such as those that appear from alignment tools such as BLAST. When are two sets of BLAST results the same? The files will not be byte identical, even with the same input parameter settings.

The Life Science Identifier (LSID) [26], along with other URI based mechanisms, are a mechanism that offers a potential solution to some of these problems [25]. LSID offers a URI based mechanism by which individual entities can be given unique identifiers. myGrid uses its own LSID authority to identify its own data holdings, but the management of external identity remains something of a problem in bioinformatics [25].

**Life cycle of an In Silico experiment**

So far we have reviewed provenance from the point of view of the experiment as a single thing and then only as a single experiment. Any experiment exists in a much wider context and all of that context could legitimately be captured as provenance. Figure 4 shows myGrid’s view of the life-cycle of an *in silico* experiment [3]. The stages in this life-cycle are:

**Experiment design**
Experiment specifications; notes describing experimental objectives and hypotheses; the databases and analytical tools to use; relevant publications and web pages.

**Experiment running**
Records for monitoring and ‘debugging’ experiments; instances of services actually used; steers of simulations.

**Experiment publication**
Data results; records linking data inputs, configurations and outcomes with experiment runs.

**Experiment knowledge discovery**
Interpretations of outcomes; the analytical processes undertaken over outcomes of collections of experiment runs; predictions and hypotheses to test in the wet lab (that is, *in vivo* and *in vitro* experiments; dry experiments are those where only computers are used).

![Figure 4: The stages of an in silico experiment.](https://example.com/fig4.png)

The cycle does not necessarily start at any one place. A previously used experiment design could be discovered, modified etc; another user might start at the publishing stage to motivate another experiment.
Provenance can be collected or used at each stage of the experiment life-cycle. During design, an experimenter will want to record design decisions, hypotheses, etc. This is collecting external provenance from the user. Other provenance might be used at this stage of an experiment: which services perform best; which services were used last time I did a similar experiment; which data were used last time I did this experiment; how did I do the experiment to get a certain type of data, etc. Obviously provenance data are collected as an experiment runs and these data are useful in debugging an experiment and in statistical analysis across organisations. Provenance is vital for publication, both of the experiment in literature and as a dissemination route itself. Publication requires materials and methods. Finally and perhaps most importantly, provenance data are needed for knowledge discovery. This includes interpretation of one experiment; discovery across experiments; design of an experiment, etc.

SYSTEMS IMPLEMENTING PROVENANCE

We have described provenance in fairly abstract terms and illustrated it with features of the myGrid project. myGrid is, however, not the only in silico experimentation system that collects provenance information, though it is perhaps the one most strongly associated with bioinformatics. Most systems collecting provenance are orientated about workflows. In the provenance survey performed as part of the ‘Provenance Challenge’ (http://dannyayers.com/2006/08/01/provenance-challenge), all systems collecting provenance were workflow systems, bar two that were embedded in operating systems. These latter systems are logging systems and different in nature from in silico experimentation and so are not relevant here.

We simply list a few systems that support provenance and direct the reader to their references for further information:

- Kepler is a scientific workflow system [14] that has grown out of the physics domain and has bioinformatics use cases.
- Chimera [16] is a workflow from the physics domain system that collects provenance.
- PASOA (as described in ‘Models of provenance’ section) is a general architecture and protocol for collecting provenance. It is a workflow system that aims to work across scientific domains and emphasises scalability.
- CombeChem is an e-Science project that uses Semantic Web technology, in particular RDF and RDF triple stores, to describe and preserve the whole process of generating chemical knowledge [27].
- Data to Knowledge (D2K) system (http://alg.ncsa.uiuc.edu/do/tools/d2k), and the CyberIntegrator system [28]. The CyberIntegrator system [28]. D2K allows for composition of data mining methods into applications and CyberIntegrator works with collections of heterogeneous workflows to perform experiments.

The models of provenance described in ‘Models of provenance’ section help us look at some of the differences between our exemplar myGrid system and some of the others. Using the pyramid model in Figure 2, we can see the differences as:

(1) At the process level, myGrid records the events occurring during a workflow run, but not the causal flow of these events, such as the invocation of one service being dependent on the completion of another service. The causal flow is recorded in other provenance systems, such as PASOA [29].

(2) At the data level, myGrid collects the provenance of data collection types (lists of protein sequence entries, etc.) and currently myGrid is the only system supporting this mechanism.

(3) At the organisation level, myGrid records the organisational metadata to state who owns a workflow or a data product, who launched a workflow run, etc. This ownership metadata is similarly provided by CombeChem [30]. Richer organisational metadata has been provided by other systems. For example, Kepler [14] builds an evolution graph of workflows to trace how a workflow is modified from another by the scientists. myGrid only has a relationship between an older version of a workflow and a current version. Kepler traces how a workflow is modified, derived from another workflow or from another version of the workflow. This results in a tree of workflows, similar to the data derivation graph seen in Figure 3.

CombeChem integrates bibliographic information about chemical data with their provenance. At the knowledge level, myGrid enables recording of both the user-specified and the externally published knowledge provenance. Some systems support user-specified annotations, e.g. NCSA, Mindswap, ISI, etc. CombeChem integrates the external knowledge provenance, such as the chemical name of a data product, or the chemical species to which it belongs, etc. None of the existing systems, however, provide both types of knowledge level provenance.

If we look at the internal and external dimension of provenance (see ‘Models of provenance’ section), the EU Provenance project gives the external logs, but myGrid does not support such external provenance delivered by actors such as third party web services. This is largely due to the fact that bioinformatics services do not expose their provenance information (such as the CPU time costing them to produce a data product) unless they have reached an agreement with the provenance system (such as the provenance protocol provided in PASOA). myGrid integrates external knowledge provenance by annotations from the user.

This non-exhaustive survey is simply a starting point for reading about support for provenance in in silico experimentation. The main thrust is from workflows. All have a basis in collecting what myGrid and others call process and data provenance. The collection of organisation and knowledge provenance is somewhat more variable. Semantic Web technologies such as RDF are popular for representing provenance graphs for the obvious reasons that it is graph based and has a semantic component.

One danger, familiar to bioinformatics, is the possibility of a set of mutually incompatible and heterogeneous provenance systems. In order to avoid this and to spread best practice within provenance work, the community has instigated the Provenance Challenge (http://dannyayers.com/2006/08/01/provenance-challenge/). The first challenge (First Provenance Challenge: http://twiki.ipaw.info/bin/view/Challenge/) aimed to gather an understanding of:

- The representations that systems use to document details of processes that have occurred;
- The capabilities of each system in answering provenance-related queries;
- What each system considers being within scope of the topic of provenance (regardless of whether the system can yet achieve all problems in that scope).

The first challenge aimed to explore the scope and boundaries of provenance. The second challenge (Second Provenance Challenge: http://twiki.ipaw.info/bin/view/Challenge/) to be held in 2007 will focus on interoperability. In the first, each system had to run a workflow involved in brain imaging and then answer a set of queries. In this second challenge, a system has to demonstrate interoperability by taking the provenance of another system and transform it to their system and perform some queries.

The existence of this challenge sends out two important messages. First is that there is a lot of interest in providing provenance support for in silico experiments across science. The second is that there is recognition of the breadth of possibilities in provenance and the opportunity to learn from other’s provenance support. Most importantly, the ability to interoperate makes this important knowledge resource more manageable.

DISCUSSION

In this briefing, we have introduced the field of knowledge management for recording the information about the running of in silico experiments. This area of knowledge management goes under the name of provenance. The eScience programme in particular has started developing the notion of provenance into a set of models that can be applied across many applications in many domains. So far, workflow systems have led the way in offering provenance support, but there is no reason why other applications should not also do so.

Several factors have contributed to the emergence of provenance as a feature of these systems. Firstly, workflow systems are a generic way of designing and running in silico experiments. This means one-off, bespoke recording solutions are not needed; it becomes cost effective to adopt provenance procedures. Secondly, the eScience context envisions large-scale in silico experimentation. Thirdly, and
perhaps most importantly, the realisation of the need to apply some scientific rigour to the management of in silico experiments.

All aspects of an experiment, its design, data inputs and outputs, intermediate values, etc. are all knowledge of and about that experiment. As in silico experiments are, by their nature, performed using computers, it is feasible to collect much of this information automatically. This gives an opportunity for completeness and consistency often lacking in human records that are often not well kept (if at all) for many bioinformatics analyses.

Capturing this knowledge allows a great many kinds of provenance dependent questions to be supported. Perhaps the primary activity supported by provenance is validation and verification of individual experiments and discovery across multiple experiments.

It should be noted, however, that the idea of provenance is not new. The advent of eScience has meant that the infrastructure has been developed to support it. Laboratory scientists have always recorded experiment provenance, but not by that name. While scientific rigour might be used in the design and interpretation of in silico bioinformatics experiments, such rigour has not been applied to the recording of those experiments. Provenance means that this rigour is now possible.

One interesting future possibility is the use of provenance to develop the ‘minimal information for the annotation of a bioinformatics experiment’. Such annotation minima have become popular in bioinformatics [33], but largely for wet experiments. If nothing else, provenance for dry, in silico experiments should contain enough information to re-create an experiment. It should also allow an experiment to be validated and verified by various parties involved in using the experiment and its data holdings. Provenance should contain the ‘minimal information for the annotation of a bioinformatics experiment’.

Knowledge of and about an in silico experiment is a significant knowledge management issue in bioinformatics. The past few years have seen it become an issue as bioinformatics analyses become larger scale and more complex. Support for provenance will provide the basis for applying scientific best practice across the spectrum of bioinformatics analyses.

Key Points
- Recording what has happened during an experiment, how it was done, why it was done, etc. This is a knowledge management issue.
- Recording the knowledge of how an experiment was done and why it was done is an experiment’s provenance.
- Provenance recording is valuable for scientific best practice. Explicit recordings of the ‘materials and methods’ of an in silico experiment is a knowledge resource in its own right that can be a valuable source of data.
- Generic models of what provenance to record and how to record that provenance are emerging from eScience research. This eScience provenance can be described as either external or internal. Each of these can also be process, data, organisational or knowledge provenance.

Acknowledgements
The authors would like to acknowledge the assistance of the whole myGrid consortium. J.Z’s work is supported by a studentship from the UK e-Science programme EPSRC grant GR/R67743.

References