BIOINFORMATICS

Vol. 24 ISMB 2008, pages i304–i312
doi:10.1093/bioinformatics/btn157

Towards the use of argumentation in bioinformatics: a gene expression case study

Kenneth McLeod[1,*] and Albert Burger[1,2]

[1]Department of Computer Science, Heriot-Watt University and [2]MRC Human Genetics Unit, Edinburgh, UK

ABSTRACT

Motivation: Due to different experimental setups and various interpretations of results, the data contained in online bioinformatics resources can be inconsistent, therefore, making it more difficult for users of these resources to assess the suitability and correctness of the answers to their queries. This work investigates the role of argumentation systems to help users evaluate such answers. More specifically, it looks closely at a gene expression case study, creating an appropriate representation of the underlying data and series of rules that are used by a third-party argumentation engine to reason over the query results provided by the mouse gene expression database EMAGE.

Results: A prototype using the ASPIC argumentation engine has been implemented and a preliminary evaluation carried out. This evaluation suggested that argumentation can be used to deal with inconsistent data in biological resources.

Availability: The ASPIC argumentation engine is available from http://www.argumentation.org. EMAGE gene expression data can be obtained from http://genex.hgu.mrc.ac.uk. The argumentation rules for the gene expression example are available from the lead author upon request.

Contact: kcm1@hw.ac.uk

1 INTRODUCTION

Biologists have access to an ever increasing number and range of online data resources (Bateman, 2007). Many of these resources contain inconsistent data. This is not surprising as biology is a complex science in which countless parameters affect the outcome of every experiment. Added to this is the human element that causes two identical results to be evaluated differently by different people. The consequence is that two seemingly identical experiments can produce contradictory outcomes. These experiments may be stored in one or more of the online resources that service a particular field.

If both of these experiments are published by the same resource, it becomes inconsistent. However, if each experiment is published by a different resource, then the inconsistency is between resources and becomes harder to detect. Regardless of where it occurs, inconsistency confuses users, forcing them to research further in order to answer their query.

In McLeod and Burger (2007) it was suggested that argumentation could be one solution to this problem. By using all the resources in a field, arguments could be created for and against potential answers to a query. These arguments could be presented to the user, providing them with a powerful set of knowledge that could be used to identify the most likely solution to the query.

This case study created a prototype using a third-party argumentation engine from ASPIC, Argumentation Services Platform with Integrated Components (www.argumentation.org), to generate arguments for the data held in the EMAGE developmental mouse gene expression database (Davidson *et al.*, 1997). Future work will extend this to include data from a complementary developmental mouse gene expression database, GXD (Ringwald *et al.*, 2001).

Section 2 starts with a discussion of argumentation. It is followed in Section 3 by an examination of the gene expression resources EMAGE and GXD in order to explain the need for argumentation. In Section 4, the argumentation engine is introduced, and Section 5 describes how the knowledge in the domain was interpreted for use with the ASPIC argumentation engine. Subsequently, in Section 6, the creation of arguments by ASPIC is discussed. The study continues with a preliminary evaluation of the prototype in Section 7 and a discussion of the work in Section 8 before the conclusions are presented in Section 9.

2 ARGUMENTATION

An argument is a reason to believe that something is true. Arguments can be used to support or attack statements. Argumentation (Carbogim *et al.*, 2000; Pollock, 2002) is the use of computers in the process of arguing, either for helping humans to argue or by actually using the computers to conduct the argument. As an approach argumentation mimics a human process and appears intuitive to human users (Williams and Williamson, 2006).

The actual form of an argument will depend on the theory being implemented. Commonly, an argument is viewed as being a series of inference rules that are chained together in a manner similar to logic programming: there are a number of statements (premises) that if true, imply that the conclusion is true. A premise of a rule may be satisfied by the conclusion of another rule. So in order for the first rule to be satisfied, all the premises of the second rule must also be satisfied. Eventually, premises will be satisfied because they are known to be true: they appear in a knowledge base that holds all currently accepted knowledge for the domain. As the knowledge changes, new arguments will be formed. These new arguments may contradict existing arguments, thus creating conflict.

Conflict between arguments is usually represented in two ways. The first is rebuttal, where two arguments have opposite conclusions: e.g. *it is raining outside* versus *it is not raining outside*. Undercut is the second form of conflict. It is an attempt to show that another argument is not valid because the premises do not imply the conclusion. For example, an argument that someone will get wet

*To whom correspondence should be addressed.

because it is raining outside is undercut by the knowledge that the person has an umbrella.

Argumentation provides a means of resolving this conflict. The arguments can be weighed and compared, with the strongest argument(s) winning. Thus, the conclusion supported by the strongest argument(s) wins.

Argumentation has been used in areas such as medicine, law (Bench-Capon and Prakken, 2006) and practical reasoning (Rahwan and Amgoud, 2006). Medical uses of argumentation vary from systems that provide advice on administering drugs (Hurt *et al.*, 2003), to those that help clinicians plan the management of chronic illness through the provision of decision support (Glasspool *et al.*, 2006). Argumentation has also been used to generate explanations of diagnosis, produced by other computational means, for the benefit of patients (Williams and Williamson, 2006). In contrast to medical-informatics, bioinformatics has produced little work on argumentation, although Jefferys *et al.* (2006) used argumentation to successfully evaluate the output of a protein prediction tool. This work showed clearly that argumentation could be applied to bioinformatics tools, but what about bioinformatics data resources?

3 ON THE NEED FOR ARGUMENTION IN BIOLOGY

Bioinformaticians have access to an ever-increasing range of online resources (Bateman, 2007), many of which publish experimental results for a particular field. For example, the results of *in situ* gene expression experiments for the developmental mouse are published in both EMAGE (Davidson *et al.*, 1997) and GXD (Ringwald *et al.*, 2001). Genes are a set of instructions that tell the body what to build, e.g. a particular set of genes results in the creation of a nose and a different set of genes produces whiskers. *In situ* gene expression experiments are designed to identify the genes that are active in a particular anatomical structure. For that structure, the experiment sets out to determine if the gene is active (expressed) or not active (not expressed). EMAGE and GXD take their knowledge of embryonic anatomical structures from a common anatomy, EMAP (Baldock and Davidson, 2008), though GXD has additional structures for the adult mouse.

An *in situ* gene expression database, such as EMAGE or GXD, allows its users to find the conclusions of gene expression experiments. These conclusions link a gene to a structure, with a level of expression (i.e. *expressed* or *not expressed*). The database also provides provenance data such as: who the research team was, details of where the experiment was published, the images showing the experimental result and details of the mouse experimented on. When using such a resource, the user will start by asking for the genes (not) expressed in a particular structure, or for the list of structures where a specific gene is expressed.

The complex nature of biology means that it is possible for experiments to produce conclusions that seem to be contradictory, e.g. one experiment may suggest the gene *Hoxb1* is expressed in the Neural Ectoderm (EMAP:151) and a second report that it is not. There are many reasons why this could be the case. For example, the experiments though very similar may be slightly different, e.g. using different probes may have produced different results, the results may have been analysed differently, e.g. different interpretations of the original gene expression images generate different experimental conclusions, and there is always the possibility of a genuine error, e.g. when entering the data into the database.

In addition to internal inconsistencies, resources covering the same field may contradict each other. For example, although EMAGE and GXD have a high level of duplication (in terms of data), their contents are not identical. To illustrate this, consider the gene *Bmp4* and the structure Future Brain (EMAP:1199). At the time of writing this article, GXD contains only one experiment for this combination, and it suggests that *Bmp4* is not expressed. EMAGE has this experiment, but in addition it contains another three experiments, all of which indicate that *Bmp4* is expressed in the Future Brain. With all the available evidence, the most likely conclusion is that the gene is expressed; however, if the user relies on a single resource, in this case GXD, a wrong conclusion may be drawn.

Because these resources are incomplete, it is vital that they are both used, in order to generate as many arguments as possible. However, this highlights a number of issues, both practical and theoretical, which require consideration. An example of a practical issue would be in identifying experimental results that are duplicated in the other resource. If an argument is created from data in EMAGE, there is no point in creating an argument based on the same data in GXD. Theoretical issues include determining whether or not the knowledge used to create arguments for EMAGE can be successfully applied to a similar resource such as GXD. These issues are not considered in this study but will be the subject of future work.

Regardless of location or reason, contradictions are confusing for users, and require them to investigate the data more fully, often to the extent of re-reading the original paper in which the result was published. It would be useful to conduct an investigation of the data automatically, presenting the findings to the user in a manner that they could analyse easily. It is hoped that argumentation may provide a mechanism to achieve this.

4 ARGUMENTATION ENGINE

Many different types of argumentation software exist. Some are used for visualization and explanation of arguments, e.g. Araucaria (Reed and Rowe, 2001), some for decision support (Fox, 2001) and some for collaborative decision support (Gordon and Karacapilidis, 1997). However, an inference tool that generates and evaluates arguments is used in this study.

This case study intended to create a web-based tool that dynamically pulled data on-demand from EMAGE to conduct argumentation. For this reason, it required a robust inference tool that could be integrated readily into applications.

Few tools are available that meet these requirements. Many different theories for argumentation have been proposed, but few have a robust implementation that can be integrated freely into another application. For example, Gordon (1993) produced an implementation of his theory, but it is not available publicly. Oscar is an implementation from Pollock (2002) which is available for download. Unfortunately, it is programmed in LISP making it difficult to integrate. The original argumentation engine concept and theory was produced by Dung (1995), but it did not include an implementation.

ASPIC had the goal of standardizing argumentation theory in Artificial Intelligence and developing a suite of tools that could be used in standard application areas such as dialogue, decision-making

and machine learning. The foundation of this implementation is a JAVA tool that generates arguments using inference. ASPIC's argumentation engine is still in development. Consequently, it is not as robust as might be desired, though it is reliable enough to work for more complex examples than those presented here. Crucially, the engine's design ensures that it can be integrated into other projects. Although the code is not open source, the engine is available publicly.

The theory behind the engine is based on the work of Dung (1995). Dung's system is abstract, in the sense that the notion of an argument was not defined. ASPIC, however, defines an individual argument in the form of an inference tree: inference rules are chained together to form an argument that is organized in a tree structure (Fig. 1). The sole form of attack in the system is rebuttal. ASPIC mimics an undercut by allowing the user to assign a name to a rule and then create a second rule that rebuts the name (Fig. 2). This succeeds because the name is automatically treated as a premise to its rule, and therefore the second rule rebuts a premise of the first rule.

Input to the engine is: a set of knowledge that models the domain being argued about, a set of rules used to infer new knowledge in the domain (when instantiated a rule forms an argument for the knowledge generated), a set of parameters that configure the behaviour of the engine, and a query that the user wishes the engine to argue about. Once a query is submitted, the engine generates arguments that support and attack the query before evaluating them. Output is the arguments, their status and the relationships between them. In terms of status, the engine records whether or not an argument is true (w.r.t current knowledge) and for relationships the engine provides a list of which arguments attack which other arguments. This information can be presented visually, in a graph, or in textual form.

ASPIC's argumentation engine can be used via a supplied Graphical User Interface (GUI), or programmatically through a JAVA Application Programming Interface (API). The engine has a fixed knowledge syntax, so an argument must conform to the specification created by the designers. When using the GUI, input to the engine has the form of first-order logic. The chosen logic is similar to PROLOG (Bratko, 2000) and features weak and strong negation. The JAVA API is designed around the logic, with the methods reflecting the underlying language by using terminology such as Variable, Term, Consequent and Antecedent. It is the API that the rest of this article deals with.

```
Conclusion
  Premise 1
        Premise 3
        Rule: Premise 1 <- Premise 3
  Premise 2
  Rule 1: Conclusion <- Premise 1 & Premise 2
```

Fig. 1. Arguments in ASPIC are stored in a tree structure. The earlier argument has the conclusion *Outcome* and three contributing subarguments. *Rule 1* provides the inference rule used to reach the conclusion. This inference rule states that *Outcome* is true if both *Premise 1* and *Premise 2* are true. *Premise 2* is known to be true. *Premise 1* is the conclusion of another argument, and is only true when *Premise 3 is true*.

```
[ID_1] Conclusion <- Premise 1 & Premise 2

To undercut this rule:
~ID_1 <- Premise 3 & Premise 4
```

Fig. 2. Undercutting an argument in ASPIC. The first rule states that *Conclusion* is true when *Premise 1* and *Premise 2* are both true. This rule is assigned the name *ID_1*. The second rule states that when *Premise 3* and *Premise 4* are both true, the rule called *ID_1* cannot be applied.

5 FORMALIZATION OF KNOWLEDGE

Argumentation takes place in a particular domain. That domain could be some everyday area such as planning how to travel to London, or it could be something more specialized such as *in situ* gene expression.

The argumentation engine is given two forms of knowledge from the domain of gene expression. The first documents the current state of the domain, i.e. what is believed to be true, which in this case is the results of gene expression experiments. The second type, is the knowledge of how to interpret the first. This knowledge came from the EMAGE curator, and was converted into inference rules that the engine uses to infer new arguments.

The domain's state will change continually as new experiments are submitted to EMAGE daily. However, the knowledge of how to interpret that experimental data changes far less often. Therefore, it is safe to gather expert knowledge in advance and store it for use later. Due to the high rate of change, the experimental knowledge must be obtained when it is to be used. This on-demand creation of knowledge is achieved by pulling data through EMAGE's SOAP-based web service and subsequently converting it.

Knowledge can be strict or defeasible. Strict knowledge is definitely true, e.g. London is the capital of the UK. Defeasible knowledge may be true, but an element of doubt remains, e.g. it is raining, therefore I will get wet. Associating knowledge with a real number between 0 and 1 indicates the user's *degree of belief* in an item of knowledge. If no degree of belief is specified, the piece of knowledge is assumed to be strict (have confidence equal to 1). This confidence score is how ASPIC assigns a strength to an argument: the higher the score the stronger the argument.

In addition, each piece of knowledge can be assigned a description: a piece of natural language text that describes the knowledge. The description can hold a simple explanation of a rule or fact. It is also possible to assign a description to the conclusion of a rule. Consequently, an argument can be viewed as a series of logic or natural language statements.

5.1 Expert knowledge to inference rules

Inference rules are used by ASPIC to infer new arguments. They model the inference processes of the domain being investigated. Once captured the inference processes need to be converted into ASPIC's chosen logic for use in the engine. The example featured attempts to argue over the accuracy of data stored in EMAGE. As such, new arguments are inferred from the contents of the database according to processes suggested by the EMAGE curation team. This team is responsible for maintaining the quality of the resource by reviewing the experiments submitted for inclusion in EMAGE.

Expert knowledge was gathered in advance during a series of meetings. These meetings started with informal discussions and

moved onto using concrete examples to illustrate how the curator processes information. Although ASPIC's engine uses a first-order logic, biologists tend to prefer natural language. In order to provide a bridge between the two, the notion of an *argument schema* (Walton, 1996b) is used. This allowed the expert's reasoning to be captured in a semi-formal way using natural language.

A schema provides a natural language inference rule that documents an inference that can be assumed to be true unless shown otherwise (defeated by a counter-argument). A schema also provides a collection of *Critical Questions* that highlight exceptions to, and extra conditions on, the use of the inference rule. All the knowledge needed to create a formal logic inference rule is documented in a manner that can be easily understood by biologists.

For example, when an EMAGE curator evaluates an experiment, they record their confidence in the experiment as a score between 0 and 3, with 3 indicating a high level of confidence. The curator's confidence is made public because a high-quality experiment is more likely to produce a correct result than an experiment the curator has less confidence in. Intuitively, it can be suggested that if the curator has high confidence in the experiment, the user can have high confidence in the result of the experiment. This would lead to something like the following schema based on Walton's schema for an Expert (Walton, 1996a):

```
EMAGE is a leading resource on mouse in-situ
   gene expression
EMAGE has C confidence in experiment E
   suggesting gene G is expressed in
   structure S
Therefore we may be C confident that G is
   expressed in S
```

Assuming that anyone who uses the system automatically accepts the initial premise that EMAGE is an expert resource, it is possible to simplify the above schema and represent it in a PROLOG-like syntax (with capital letters indicating variables that unify and lower case letters indicating constants), so the basic rule is:

```
expressed(Gene, Structure) <-
 experiment(Id, Gene, Structure, expressed),
 confidence(Id, Confidence).
```

The problem with this rule is that the confidence EMAGE has in the experiment is not passed to ASPIC. There should be a direct link between EMAGE's confidence and the strength of the argument; therefore, it is necessary to add a degree of belief. In the instance of EMAGE having high confidence the argument should be strong and thus have a high degree of belief, for example 0.8. This can be set when passing the inference rule to ASPIC using its JAVA API.

A selection of further rules can be seen in Table 1. These rules use notions such as Theiler Stages, Spatial Annotation and Textual Annotation. The *Theiler Stages* are the 26 developmental phases of a mouse embryo. Each experiment must be mapped to one of these stages. The results of gene expression experiments (2D section images) can be described with respect to the EMAP anatomy ontology or spatially mapped into the 3D embryo models (one per Theiler Stage) of EMAP. These are referred to as *Textual Annotation* and *Spatial Annotation*, respectively. Rules 3, 4 and 5 from Table 1 are all variations of the schema discussed earlier in this section.

5.2 State of domain knowledge

ASPIC refers to each item of knowledge (or belief) referring to the current state of the domain as a *fact*; like inference rules these can be strict or defeasible. The EMAGE resource provides the setting for this case study, so the facts given to the argumentation engine correspond directly to the data held in EMAGE.

The contents of EMAGE can be abstracted to knowledge about an experiment and its conclusion. The conclusion is literally that a gene was (not) expressed in a particular anatomical structure. The experimental information states: who performed the experiment,

Table 1. Some of the rules defined by the EMAGE curator

ID	Description
1	If a gene G, is expressed in a structure S, in Theiler Stage $(T-1)$ and also in Stage $(T+1)$, then G is very likely to be expressed in S in Stage T.
2	If the user, after examining the image of the experimental result, is confident that the gene G, is expressed in the structure S, then G is very likely to be expressed in S.
3	If a spatial annotation *SA*, suggests a gene G is expressed in structure S, and the curator has high confidence in *SA*, then we may have *high* confidence that G is expressed in S.
4	If a textual annotation *TA*, suggests a gene G is expressed in structure S and the curator has high confidence in *TA*, then we may have *high* confidence that G is expressed in S.
5	If a textual annotation *TA*, suggests a gene G is expressed in structure S and the curator has medium confidence in *TA*, then we may have *medium* confidence that G is expressed in S.
6	If the user does not trust the research team that conducted experiment E, then all spatial and textual annotations based on that experiment should have a low level of confidence.
7	If a spatial annotation *SA* and a textual annotation *TA* disagree, then always trust *TA*.
8	If two experiments disagree on whether, or not, a gene G is expressed in structure S and the user believes the experiments are examining different parts of S, then G is likely to be expressed in part of S.
9	If two experiments disagree on whether, or not, a gene G is expressed in structure S and the user believes the experiments are examining different parts of S, then G is likely to be not expressed in part of S.

Rules 1–7 are relatively straightforward. However, Rules 8 and 9 may require further explanation. They state that if two experiments are examining different parts of the same structure both results can be correct regardless of their conclusion. For example, consider two experiments on the human hand. The first experiment may find a particular gene expressed in the thumb, and the second conclude that the same gene is not expressed in the index finger. These experiments show that the gene is both expressed and not expressed in the hand.

what type of experiment it was, how the gene was detected, what kind of mouse the experiment was performed on (its species, its age, whether it was normal or abnormal) and it provides photographs of the result taken by the researchers.

Consequently, the minimum requirement is to provide ASPIC with knowledge on genes, anatomical structures, the relationship between the two and some details of the experiment that established the relationship.

A fact is treated as a simplified inference rule, i.e. a rule without premises. One possible way of saying that an experiment in EMAGE, with the associated identifier EMAGE:772, reported the gene *Hoxb1* was expressed in the Neural Ectoderm (EMAP:151) is:

```
experiment(
'EMAGE:772', 'Hoxb1', 'EMAP:151', expressed).
```

Not all of the facts can be generated easily, due to the impossibility of automatically processing the experimental images. These images are taken by the researchers at the end of the experiment, and are stored in EMAGE as part of an experiment's provenance. Image analysis is a vital part of evaluating the quality of the result: it is done manually by the EMAGE curator. Consequently, in this study, the images are presented to the human user and they are asked specific questions such as: these images are from two experiments that examine the same structure, do they appear to investigate the same area? These questions are straightforward for a regular user of EMAGE to answer, but are more challenging for someone with less experience.

6 GENERATING ARGUMENTS

Arguments are generated from the contents of the knowledge base, in response to the user posing a query. The results of the query are returned for the user to examine.

6.1 Query

The query is the conclusion that the user wishes ASPIC to argue about. It will take the form of a fact, and will conform to the earlier discussion in all but one respect: it will not have a degree of belief associated with it. So in this case, an example would be:

```
expressed('Hoxb1', 'EMAP:151').
```

Once the query has been created its status is determined by the argumentation engine.

6.2 Evaluating a query

ASPIC uses a dialogue game to determine the status of a query (ASPIC, 2004). The knowledge given in the query can be *undefeated* (true with respect to current knowledge), *defeated* (false with respect to current knowledge), or *unknown*.

The game features two computer players, the Proponent (PRO) who attempts to prove the query, and the Opponent (OPP) who tries to stop PRO. The game starts with PRO creating an argument to support the query (an argument whose conclusion is identical to the query). This process starts by searching for a rule with an appropriate conclusion. Once found, rules with conclusions that are identical to the premises are sought. If the premises cannot be satisfied in this manner, the facts are examined to determine if they satisfy the premises.

OPP now attempts to defeat PRO's argument. To succeed, OPP's argument must rebut part of PRO's and have a higher degree of belief. OPP starts by trying to construct arguments that rebut the conclusion of PRO's argument. If that cannot be done, OPP attempts to rebut the premises.

If OPP succeeds in defeating PRO's argument, PRO will attempt to counter OPP's argument by defeating it. This process of attack and counter-attack continues until one player (PRO or OPP) fails to defeat the other's argument. If PRO is stopped, they try a new line of defence by creating a new argument, to support the conclusion that OPP has defeated, if they fail OPP wins. However, if OPP fails, they try to defeat one of PRO's previous arguments, if they cannot do so PRO wins.

For an example we shall use a simplified set of data for *Hoxb1* in EMAP:151, ignoring the distinction between textual and spatial annotations (see Section 5.1). EMAGE has two relevant experiments. The first suggests that *Hoxb1* is expressed in EMAP:151 and the second that it is not. The EMAGE curator has medium confidence in the first experiment, and a high level of confidence in the second. In the game, PRO starts by using the first experiment to create the argument in Figure 3 (based on a variation of the schema in Section 5.1), which OPP defeats by creating the argument in Figure 4, based on the second experiment (and a different variation of the schema in Section 5.1).

The next argument of PRO depends on the information provided by the user. Since it is impossible for the system to evaluate the images of experimental results the user is asked to help. They are given a number of questions to answer, for example: are the two experiments dealing with the same part of the structure? This question relates to Rules 8 and 9 from Table 1, and is asked because it is possible for a gene to be expressed in one part of an anatomical structure but not expressed in another part of it (e.g. a gene may be expressed in the index finger but not the thumb, as the index finger and thumb are two separate parts of the hand, the gene is both expressed and not expressed in the hand). If the user answers the question by suggesting that the experiments are examining different parts of EMAP:151, then it is possible that *Hoxb1* is both

```
Hoxb1 is expressed in EMAP:151
       EMAGE has an experiment suggesting Hoxb1 is expressed
       The EMAGE curator has medium confidence in the experiment
       If the curator has medium confidence in the experiment, then we may
            have medium confidence in the experiment and its result

        Degree of belief = 0.4
```

Fig. 3. Argument for *Hoxb1* being expressed in EMAP:151.

```
Hoxb1 is not expressed in EMAP:151
       EMAGE has an experiment suggesting Hoxb1 is not expressed
       The EMAGE curator has high confidence in the experiment
       If the curator has high confidence in the experiment, then we may
             have high confidence in the experiment and its result

        Degree of belief = 0.8
```

Fig. 4. A counter argument to the argument in Figure 3. Because the EMAGE curator has more confidence in the experiment used in this argument than the experiment used in Figure 3, this argument has a higher degree of belief and so defeats the argument from Figure 3.

```
Hoxb1 is expressed in EMAP:151
       EMAGE has an experiment suggesting Hoxb1 is not expressed
       EMAGE has an experiment suggesting Hoxb1 is expressed
       The experiments look at different parts of the same structure
       If the experiments look at different parts of the same structure,
           they can both be correct, so the gene is expressed.

        Degree of belief = 0.9
```

Fig. 5. A second argument, based on Rule 8 from Table 1, showing *Hoxb1* is expressed in EMAP:151.

```
Hoxb1 is not expressed in EMAP:151
       EMAGE has an experiment suggesting Hoxb1 is not expressed
       EMAGE has an experiment suggesting Hoxb1 is expressed
       The experiments look at different parts of the same structure
       If the experiments look at different parts of the same structure,
           they can both be correct, so the gene is not expressed.

        Degree of belief = 0.9
```

Fig. 6. A second argument, based on Rule 9 from Table 1, showing *Hoxb1* is not expressed in EMAP:151.

expressed and not expressed in EMAP:151. This leads PRO to produce the argument in Figure 5 by using Rule 8 from Table 1. As this argument defeats OPP's previous argument (based on the EMAGE experiment suggesting *Hoxb1* was not expressed) PRO's first argument is reinstated because it is no longer attacked. OPP must counter PRO's argument and does so with the same logic as PRO (Rule 9 from Table 1): the experiments are using different parts of EMAP:151, so *Hoxb1* can be both expressed and not expressed, and therefore it is not expressed (Fig. 6).

Currently there are two arguments of equal strength that contradict each other (Figs 5 and 6) . The outcome of this conflict depends on which type of game semantics is used. ASPIC provides two game semantics, *skeptical* and *credulous*, for the user to choose between. When a skeptical game is played, if there is any doubt about the acceptability of an argument it is rejected. In this case, there is doubt about the acceptability of both arguments, and so they are both rejected. The credulous game is implemented in such a way that even if there is doubt about one of PRO's arguments it is accepted, whilst OPP's argument is rejected if there is any doubt. So here PRO's argument that *Hoxb1* is expressed is accepted, with OPP's counter argument being defeated. It is left to the user to decide which game is most suitable for their situation.

Adopting credulous semantics, PRO's last argument is accepted. Because of this, both of OPP's arguments are defeated, leaving both of PRO's arguments undefeated. OPP must try to find another argument that defeats one of PRO's two arguments. However, there are no more arguments available, and so OPP fails. This game has been won by PRO.

PRO starts a second game with the argument from Figure 5 (as the two experiments are looking at different parts of EMAP:151 *Hoxb1* can be both expressed and not expressed, and therefore it is expressed). The same arguments as before are constructed, once again PRO wins.

PRO can construct no more arguments to support the query so the game is over. The results are given to the programmer to manipulate as they wish.

The results come in two separate parts. The first is a series of *yes* and *no*. Each one represents an argument that PRO has constructed to support the query. As PRO won both games, the results from this example are *yes* and *yes*.

The second part of the results is called the *proof*. Essentially it is all the arguments used in the game. If calculated the status of the argument is also recorded. In this example, the two arguments provided by PRO are undefeated with both of OPP's arguments being defeated. The programmer can present the arguments to the user in any way they wish. However, when communicating with biologists it makes more sense to use a natural language form similar to that used in this section (Fig. 4) by using the descriptions attached to rules and facts.

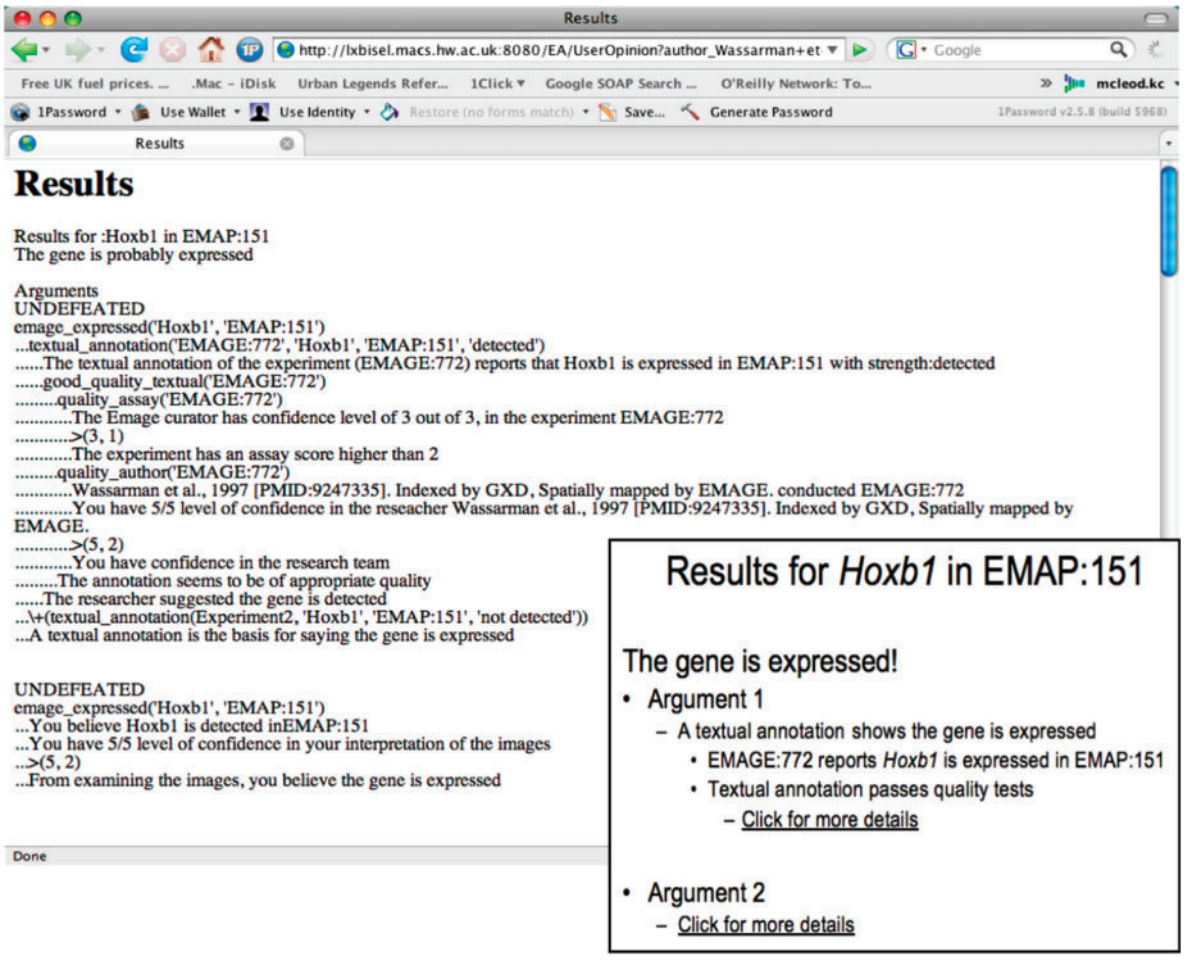

Fig. 7. Screen shot from the prototype (top-left) with simplified presentation in a mock prototype (bottom-right).

Although this evaluation of a query may seem complicated, it is essentially the type of thought-process naturally deployed by a human user. The apparent complexity relates primarily to the need to formalize this process for computational purposes. Fortunately, the details of this formalization need not be communicated to the end user and our initial evaluation (see Section 7) adds weight to our view that the underlying argumentation reasoning is accessible and helpful to biologists.

7 EVALUATION

Once the implementation of the above system was completed, a preliminary evaluation was undertaken. This informal study involved demonstrating the system to the EMAGE curator, and recording the feedback given.

Overall the system was well received. The tool was deemed easy to use, and the arguments were presented in far less time than the curator had expected. The arguments made sense to the curator, and they covered the majority of the points the curator wished to see. The curator felt that the arguments would be enough for most people to evaluate the data from EMAGE, and thus determine whether or not a gene was expressed in a particular structure. As such the system was a success.

Although feedback from the curator was positive, four issues clearly require to be tackled. The first is the presentation. The examples discussed above are simplified in order to improve clarity. However, the prototype displayed arguments in a rudimentary manner using a slightly amended version of a method ASPIC provided for the task (see top-left of Fig. 7). This resulted in a confusing output. Much of this output was redundant as it restated what had already been given. For example, in the first argument, the five lines *quality_author('EMAGE:772')* through to *You have confidence in the research team* said who the research team was twice, and that the user had confidence in this team twice. Consequently, the test user was presented with a simplified version of these arguments in a mocked-up prototype (see bottom-right of Fig. 7).

Feedback from the curator suggested simplifying further the presentation of the arguments. For example, subarguments were indented to show that they were separate from the main argument but still contributed to it. However, the curator did not understand the relationship. Instead, he suggested the information should be presented in a simple paragraph comprising two or three sentences.

The second key issue highlighted by the evaluation is *trust*. When using EMAGE, a user must *trust* the researcher to have performed and evaluated the experiment correctly, in addition the user must *trust* that all mistakes were detected and corrected by the journal that published the experiment, and finally they must *trust* the curator of EMAGE (or GXD) to have mapped correctly the researcher's findings to the EMAP ontology for inclusion in the database. For example, if the research team suggested that *Bmp4* was expressed in the presumptive infundibulum, then the curation team needed to map this structure to its equivalent in the EMAP anatomy (infundibular recess of third ventricle). These *trust* issues should have been made clear to the user by explicitly asking them if they had confidence in each of the above groups. The system did not do this.

The third issue related to the screen that asked the user for help in processing information. As mentioned in Section 5.2 the user was asked to analyse some information when the system could not do so. The curator felt that this screen presented the user with too many tasks to undertake. One possible solution would be to use the image analysis already undertaken by the authors, journal and EMAGE curation team.

The final issue raised by the curator related to GXD. The system worked with data from EMAGE. In real life, the curator would advise anyone with doubts over data in EMAGE to examine GXD (and vice versa), he felt that extending the system to include arguments based on data in GXD was vital.

The goal of this work was to assess the usefulness of argumentation in bioinformatics. Overall it was obvious that much work remained. However, it was also evident that the current prototype system was the first step on the way to a useful and interesting tool.

8 DISCUSSION

This work concentrates on two resources publishing *in situ* developmental mouse gene expression information. However, other resources that perform this function exist, for example, the Mouse Atlas of Gene Expression, MAGE (http://www.mouseatlas.org). Therefore, it would be beneficial to extend the system to include this and other related resources.

Unfortunately, this is not a simple task. MAGE uses its own ontology to describe the mouse anatomical structures. This ontology does not have a mapping to the EMAP ontology used by EMAGE and GXD. One structure in EMAP may correspond to parts of several structures in the MAGE ontology, and vice versa. Although work is progressing on a cross-linked mammalian ontology that will hopefully link EMAP to MAGE, currently there is no automatic mechanism to do this. At present this makes it impossible to use these resources together in this system.

If MAGE had used the EMAP ontology, there would be no reason why it could not be included in the system. Data from MAGE would need to be pulled and converted for use within ASPIC. Likewise, there would be a need for an evaluation of the current inference rules to determine if they could be applied to MAGE. It is probable that several extra inference rules would be required. With the new rules in place ASPIC would be able to argue as before. However, with futher knowledge at its disposal, it would be possible to create extra arguments and thus have a more complex argumentation process. Although this would be unlikely to have a significant effect in the example discussed here, it is possible that the integration of a large number of resources (or resources with a larger number of expert generated inference rules) might cause the argumentation process to run too slowly to be useful. In such a situation, it might be necessary to balance the inclusion of each resource against the usefulness of the information it provides for the arguments. Alternatively, it might prove helpful to investigate the other argumentation engines that are beginning to appear e.g. ArgKit (http://www.argkit.org).

Of course, in the context of the Internet, the argumentation workload can be distributed across more than one site. We envision domain-specific argumentation engines, e.g. one or more sites for *in situ* gene expression argumentation, that communicate with each other. Efforts are already underway to develop an Argumentation Interchange Format (Chesñevar *et al.*, 2006) to facilitate such interactions.

In addition, we note that there is a potential issue with scalability in terms of formalizing enough relevant domain knowledge for the purposes of argumentation. As with most semantics-based applications, it is unrealistic to expect that all relevant domain knowledge will be captured. However, the experience with the Semantic Web so far shows that even a 'little semantics goes a long way' (Wolstencroft *et al.*, 2005), and we believe that this applies equally to argumentation.

Argumentation has been used within this work to resolve inconsistencies across biological data resources. A variety of other mechanisms to integrate data and resolve inconsistency exist. For example, data reconciliation (a.k.a. data fusion) uses a function to turn multiple possible values into a single value, e.g. computing the average of four numbers (Motro and Rakov, 1998). A second possible mechanism would create multiple query plans for the resources, then select the best according to information quality criteria (Naumann, 1996). Our work is not an attempt to replace these mechanisms. We are not concerned with automatically resolving conflict, but instead wish to determine whether or not argumentation can enable biologists to resolve the differences themselves.

9 CONCLUSION

This case study explored the usefulness of argumentation in helping biologists work around conflicting information presented by an online biological database, in this case a developmental mouse gene expression database called EMAGE.

By investigating the reasoning processes of an EMAGE curator, a series of rules for assessing the quality of an experiment were produced. These rules were used by the ASPIC argumentation engine to generate arguments on the validity of the data provided by EMAGE. This enabled arguments for and against each experimental result to be produced and presented to the user.

Following an implementation of the system, an evaluation was undertaken with the EMAGE curator. The evaluation showed that the basic concept was correct: arguments could be used to highlight issues and help the user determine if data was valid.

However, it also stressed the importance of presenting the arguments in an appropriate manner, and here further work must be undertaken. This is not the only work needed, in particular an effort must be made to extend the system so that it can create arguments based on the data held in another developmental mouse gene expression database, GXD. Only then it will be possible to make an accurate assessment of the full worth of argumentation in a bioinformatics setting.

ACKNOWLEDGEMENT

This work could not have been completed without the support of the EMAGE curation team.

Funding: Funding by the EU projects Sealife (FP6-2006-IST-027269) and REWERSE (FP6-2006-IST- 506779) is acknowledged.

Conflict of Interest: none declared.

REFERENCES

ASPIC. (2004) Theoretical framework for argumentation, deliverable d2.1.

Baldock,R. and Davidson,D. (2008) *In Anatomy Ontologies for Bioinformatics: Principles and Practise*. The Edinburgh Mouse Atlas. Springer, Berlin, Germany.

Bateman,A. (2007) Editorial. *Nucleic Acids Research*, **35, (Database Issue)**, D1–D2.

Bench-Capon,T. and Prakken,H. (2006) *Argumentation. In Information Technology & Lawyers: Advanced technology in the legal domain, from challenges to daily routine*, Springer, Berlin/Heidelberg/New York. pp. 61–80.

Bratko,I. (2000) *PROLOG Programming for Artificial Intelligence*. Addison Wesley, Harlow, UK.

Carbogim,D.V. *et al.* (2000) Argument-based applications to knowledge engineering. *Knowl. Eng. Rev.*, **15**, 119–149.

Chesñevar,C. *et al.* (2006) Towards an argument interchange format. *Knowl. Eng. Rev.*, **21**, 293–316.

Davidson,D. *et al.* (1997) The mouse atlas and graphical gene-expression database. *Semin. Cell Dev. Biol.*, **8**, 509–517.

Dung,P.M. (1995) On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, **77**, 321–358.

Fox,J. (2001) Rags: a novel approach to computerized genetic risk assessment and decision support from pedigrees. *Methods Inform. Med.*, **4**.

Glasspool,D.W. *et al.* (2006) Argumentation in decision support for medical care planning for patients and clinicians. In Bickmore,T. and Green,N. (eds), *Proceedings of AAAI Spring Symposium Series 2006 (AAAI Technical Report SS-06-01)*, AAAI Press, California, USA.

Gordon,T.F. (1993) The pleadings game: formalizing procedural justice. In *Proceedings of the 4th International Conference on Artifical Intelligence and Law*, ACM Press, New York, NY, USA, pp. 10–19.

Gordon,T.F. and Karacapilidis,N. (1997) The zeno argumentation framework. In *Proceedings of the 6th International Conference on Artifical Intelligence and Law*, Melborne, Australia, ACM Press, pp. 10–18.

Hurt,C. *et al.* (2003) Computerised advice on drug dosage decisions in childhood leukaemia: a method and a safety strategy. In Dojat,M., Keravnou,E. and Barahona,P. (eds), *Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe*. Berlin, Germany, Springer, pp. 158–163.

Jefferys,B.R. *et al.* (2006) Capturing expert knowledge with argumentation: a case study in bioinformatics. *Bioinformatics*, **22**, 923–933.

McLeod,K. and Burger,A. (2007) Using argumentation to tackle inconsistency and incompleteness in online distributed life science resources. In Aes,N.G. and Isaís,P. (eds), *Proceedings of IADIS International Conference Applied Computing 2007*, Salamanca, Spain. IADIS Press, pp. 489–492.

Motro,A. and Rakov,I. (1998) Estimating the quality of databases. In *Proceedings of the 3rd International Conference on Flexible Query Answering Systems*, Roskilde, Denmark, Springer, pp. 298–307.

Naumann,F. (1996) *Quality-Driven Query Answering for Integrated Information Systems. Lecture Notes in Computer Science, vol. 2261*. Springer, Berlin, Germany.

Pollock,J.L. (2002) Defeasible reasoning with variable degrees of justification. *Artif. Intell.*, **1–2**, 232–282.

Rahwan,I. and Amgoud,L. (2006) An argumentation-based approach for practical reasoning. In *Proceedings of the 5th International Conference on Autonomous Agents and Multiagent Systems*, Hakodate, Japan, ACM Press, pp. 347–354.

Reed,C. and Rowe,G. (2001) Araucaria: software for puzzles in argument diagramming and XML. *Technical Report*, Department of Applied Computing, University of Dundee, Dundee, Scotland, UK.

Ringwald,M. *et al.* (2001) The mouse gene expression database (GXD). *Nucleic Acids Res.*, **29**, 98–101.

Walton,D. (1996a) *Appeal to Expert Opinion : Arguments from Authority*. Penn State University Press, PA. USA.

Walton,D. (1996b) *Argumentation Schemes for Presumptive Reasoning (Studies in Argumentation Series)*. Lawrence Erlbaum Associates, Mawah, NJ, USA.

Williams,M. and Williamson,J. (2006) Combining argumentation and Bayesian nets for breast cancer prognosis. *J. Log. Lang. Inform.*, **15**, 155–178.

Wolstencroft,K. *et al.* (2005) A little semantic web goes a long way in biology. In *Proceedings of the 4th International Semantic Web Conference*. Galway, Ireland, pp. 786–800.