A knowledge base for *D. melanogaster* gene interactions involved in pattern formation

Jérôme Euzenat[+,#], Christophe Chemla[*,#] and Bernard Jacq[*]

[+] INRIA Rhône-Alpes
655 avenue de l'Europe,
38330 Montbonnot Saint-Martin, France
Jerome.Euzenat@inrialpes.fr

[*] Laboratoire de Génétique et Physiologie du Développement
Parc scientifique de Luminy, CNRS Case 907,
13288 Marseille cedex 9, France
jacq@lgpd.univ-mrs.fr

Abstract

The understanding of pattern formation in Drosophila requires the handling of the many genetic and molecular interactions which occur between developmental genes. For that purpose, a knowledge base (KNIFE) has been developed in order to structure and manipulate the interaction data. KNIFE contains data about interactions published in the literature and gathered from various databases. These data are structured in an object knowledge representation system into various interrelated entities. KNIFE can be browsed through a WWW interface in order to select, classify and examine the objects and their references in other bases. It also provides specialised biological tools such as interaction network manipulation and diagnosis of missing interactions.

We are interested in the biological process of pattern formation in Drosophila and in understanding the basis of specific identity acquisition by the different body parts [Fasano *et al.* 1991; Röder, Vola and Kerridge 1992; Alexandre *et al.* 1996]. In Drosophila, different classes of genes involved in the segmentation processes (maternal, gap, pair-rule and segment polarity genes) divide the embryo along the antero-posterior axis into repeated homologous units [Nüsslein-Volhard and Wieschaus 1980; Gaul and Jäckle 1990] which will develop specific identities and morphogenetic features under the control of homeotic genes [Lewis 1978]. Specific interactions within and between these gene families are essential for the establishment of a correct body pattern. Being able to access, query and manipulate the data on these developmental genes and their functional interactions within specific regulatory networks is now an important requirement for developmental and molecular biologists studying gene regulation.

Gene molecular interactions, i.e. direct molecular interactions involving DNA, RNA and proteins, play an essential role in all known biological processes. Although different databases exist for each of these three types of macromolecules, data concerning precise molecular interactions between them are underrepresented in these databases. The majority of these databases can be classified as "mainly structural" because the core of their informational content is based on various aspects of DNA, RNA or protein sequence and/or structure. Relatively few databases have a content and an organisation which are oriented towards the biological function of the genes and the relationships between structure and function. In the field of genetic diseases, OMIM, a catalogue of human genes and genetic disorders [OMIM 1996] provides the user with both structural (e.g. molecular genetics, biochemistry, genetic mapping) and functional data (e.g. clinical features, diagnosis, inheritance) on human genes.

In order to cope with problems of genetic regulation, it is first essential to be able to describe and organise biological facts such as those described above in a standardised way. For this purpose, we recently described [Jacq *et al.* 1997] the concepts, organisation, content and use of GIF-DB (Gene Interactions in the Fly Database), a new WWW repository for data on gene interactions involved in Drosophila embryonic development and the regulatory networks in which they are involved. GIF-DB is a collection of hypertext files, each of them describing an interaction between two molecular partners (Protein/DNA, Protein/RNA, Protein/Protein). All data found in GIF-DB come from the literature.

The production of GIF-DB is a first step in the process of managing scientific information concerning genetic interactions. Although quite complete and simple to use, a collection of hypertext files is only imperfectly suited to represent and query the knowledge we already have on such complex biological problems as molecular interactions.

A further step in that direction requires a system providing more structure (e.g. objects and class hierarchies) and manipulation capabilities (e.g. classification, network

---

# Both authors contributed equally to this work.

traversing) than a relational database or a flat WWW server. Such an organisation will provide the opportunity to gather interactions into networks and to compare, check and simulate these networks.
The KNIFE system (Knowledge on Networks of Interactions in the Fly Embryo)[1], which is presented here, is a knowledge base putting together data and methods about interactions gathered in other databases. It has been developed using the TROPES knowledge representation system, is directly accessible from the WWW and provides several specific utilities such as network traversing and analysis.
The choice of an object-based knowledge representation system has already been made by several teams working on metabolism [Karp and Mavrovouniotis 1994] or genetic regulation [Perrière *et al.* 1993; Hoogland and Biémont 1997]. The more closely related work to this one is the work around the ECOCYC knowledge base on *Escherichia coli* genes and metabolism [Karp *et al.* 1996]. It is an example of a knowledge base which integrates functional aspects since one can find both data on gene structure and their function in the regulation of biochemical pathways. Although it focuses on metabolism instead of genetic regulation, there are several features common with the work presented here: use of knowledge base technology; availability through WWW, including for graph drawing; development of specific algorithms for using the networks (pathways). An important difference is that the context in which genetic interactions are valid is not well known yet.

The TROPES object-based knowledge representation system is first described. Then, the objects, modelling KNIFE data, are presented at length. The functions offered by KNIFE are exposed in the remainder: first the general functions available in TROPES (WWW browsing, classification, filtering) are presented; then, specific methods tied to the exploration of genetic interactions (graph traversing and diagnosis of missing interactions) are detailled. The discussion focuses on a long term goal: the simulation of interaction networks.

Short presentation of TROPES

TROPES is an object-based representation system which favours classification. It is here presented through its basic notions, while more specific descriptions will be found in the remaining sections.

Objects and concepts

In TROPES [Mariño *et al.* 1990; Sherpa 1995] individuals are represented as objects. The objects are partitioned into *concepts* (an object is an instance of one and only one concept). As an example, the `protein` concept concerns all the individual proteins. Ontological prerogatives are attached to the concept. They warrant the integrity of an object (i.e. that the object cannot be modified in a way which would lead it to no longer be an instance of the concept) and its identity (i.e. it can always be uniquely identified). These prerogatives play an important role when the object is created and registered.
The concept also defines the structure of its instances. The structure of an object is uniquely determined by a set of *fields* and their basic domains independently of the classes to which the object can be attached. For example, the instances of the `protein` concept have a `name`, a `size` and a `sequence` field. The basic domain of a field is either a primitive type (string for the `name`, integer for the `size`), a concept (`protein-sequence` for `sequence`) or a type constructed from primitive types and concepts with the help of set and list constructors.
The field values are part of the objects and do not depend upon the classes to which the objects are attached. For instance, the protein BICOID is represented by an object instance of the `protein` class. Its `name` field has the string `"BICOID"` for value, its `size` field has integer `494` for value and its `sequence` field has the instance of the `protein-sequence` concept also named `BICOID` for value.

Viewpoints

Objects can be seen under several *viewpoints*. For instance, `BICOID` is classified as a `transcription-factor` under the `biochemical-function` viewpoint and as a `nuclei` protein under the `initial-` and `final-sub-cellular-location`. The viewpoints allow to restrict the view on instances and to organise the concept into particular taxonomies. A viewpoint determines:

- The set of fields which are relevant under the viewpoint (the `sequence` is not relevant from the `sub-cellular-location` viewpoints as well as the `size` under the `biochemical-function` one, and thus the corresponding fields are hidden under the respective viewpoints).
- A hierarchy of classes under which the instances of the concept can be classified. Each viewpoint offers to the user a new taxonomy under which the classification operation depends on different criteria. They allow to focus on particular aspects of objects without being disturbed by others. Classes are related through the specialisation relation and determine progressive subsets of the set of instances of the concept. Under the `biochemical-function` viewpoint, `proteins` are divided into `enzymes`, `DNA-associated`, `growth-factor`, and other classes; `DNA-associated` can be

[1] `http://www-biol.univ-mrs.fr/~lgpd/knife.html`

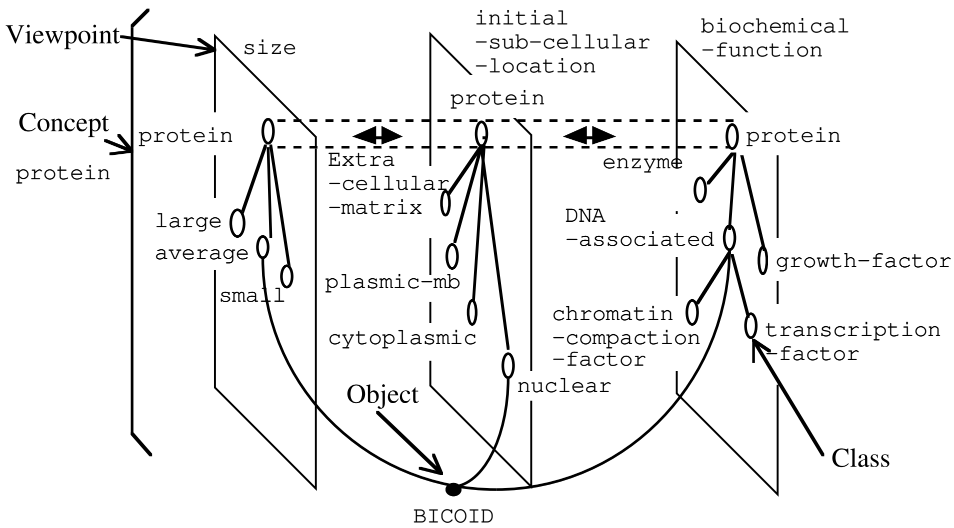

Figure 1. The `protein` concept is visible under the `size`, `initial-sub-cellular-location` and `biochemical-function` viewpoints. Each of them determines a hierarchy of classes whose root class is named `protein`. For instance, under the `biochemical-function` viewpoint there is a decomposition of the set of `proteins` following their functions. The `BICOID` object is attached to the `average` class under the `size` viewpoint, to the `nuclear` class under the `initial-sub-cellular-location` viewpoint and to the `DNA-associated` class under the `biochemical-function` viewpoint.

divided into `transcription-factor` and `chromatin-compaction-factor`.

Classes and taxonomy

A *class* defines constraints that objects must satisfy in order to belong to the class. It is a projection of the structure of the concept retaining only relevant fields and a restriction of the possible values of these fields. This is achieved with the help of:

- Primitive domain restriction provided by domain enumeration, exclusion or bounding (the `effect` field of an `interaction` has a string value enumerated by `"activation"`, `"repression"` and `"activation-and-repression"`; the `size` of an `average protein` can be between `300` and `1000` amino-acids).
- Attachment restrictions for concepts: the field values must not only be instances of a particular concept but can be constrained to belong to particular classes of that concept (the value of the `sequence` field of a `protein` must be member of the `protein-sequence` class).
- Constraints on field values. These constraints are membership constraints or constraints between fields (the `size` of a `RNA` cannot exceed that of its `DNA-sequence`).
- Cardinality restriction on sets (resp. lists) by bounding their number of elements.

An object is attached to (to be opposed to “is member of”) only one more specific sub-class under a viewpoint, but is member of all the classes of which this class is a specialisation. The interpretation of the specialisation relation is twofold: first, the members of a class are members of its super-classes (extension inclusion property); second, the constraints defined in a class apply to all the members of that class (and thus to the objects attached to all of the more specific classes). In the above example, it means that:

- all `transcription-factors` are `DNA-associated`, and
- all `transcription-factors`, as `DNA-associated`, inherit their constraints (e.g. being located in the nucleus).

So the strengthening of the constraints from class to sub-classes is parallel to the restriction of the extension. As opposed to instanciation, objects can be attached to a class and can be detached from it at anytime. Classes also allow the expression of hypothetical knowledge in terms of default values or default inference methods.

TROPES has other features which are not relevant here. Some of them will be presented when needed in the remaining sections.

Biological knowledge representation

The core of the system is described hereafter. It is made of a repository (expressed in TROPES) of concepts, classes and instances representing biological entities. The data originates from other standard repositories and GIF-DB. Figures given here are those at the date of December 1st, 1996.

General overview

The knowledge base, while still incomplete, has been carefully designed in order to reflect the complexity of gene interaction. The KNIFE base contains 16 concepts related with many different aspects of interactions. Figure 2 presents these concepts and the relationships between them through fields referring to each others. The main concepts, with regard to the present paper, are `network`, `interaction`, `expression`, `gene` and `protein`. They will be detailed in depth below.

The other concepts, although being useful to the biologist looking for where, how and why an interaction can happen, are not yet used by the automatic facilities of KNIFE. They can be briefly described in the following manner (numbers in parentheses indicate the number of instances in the actual knowledge base):

`mutant` (0): the mutant concept describes the kind of mutations which can be applied to a gene and the resulting phenotype. It is not presently used.

`binding-site-feature` (43): describes the binding-sites involved in the interaction between biological objects. They refer the actual sequence through their `sequence` field and the binding protein through the `overlaps-with-protein` field.

`precursor-rna` (27): The precursors of a gene are the RNA products which have not been spliced. It was necessary to introduce precursors because the same precursor can be spliced differently or two different precursors can give rise to the same product after splicing.

`rna` (29): `rna` objects record the information concerning RNA. An obvious one is the `sequence` field referring to the `rna-sequence` itself, the `product` field referring to the `protein` expressed by the RNA strand. The `expression` field is meant to describe in which context this particular RNA strand is to be expressed. These `rna` objects are referred to by the `gene` concept, allowing them to determine their corresponding expression patterns.

`dna-sequence` (51), `rna-sequence` (4), `protein-sequence` (25): the individual sequences of DNA are described by subsequences (called `dna-part`) and

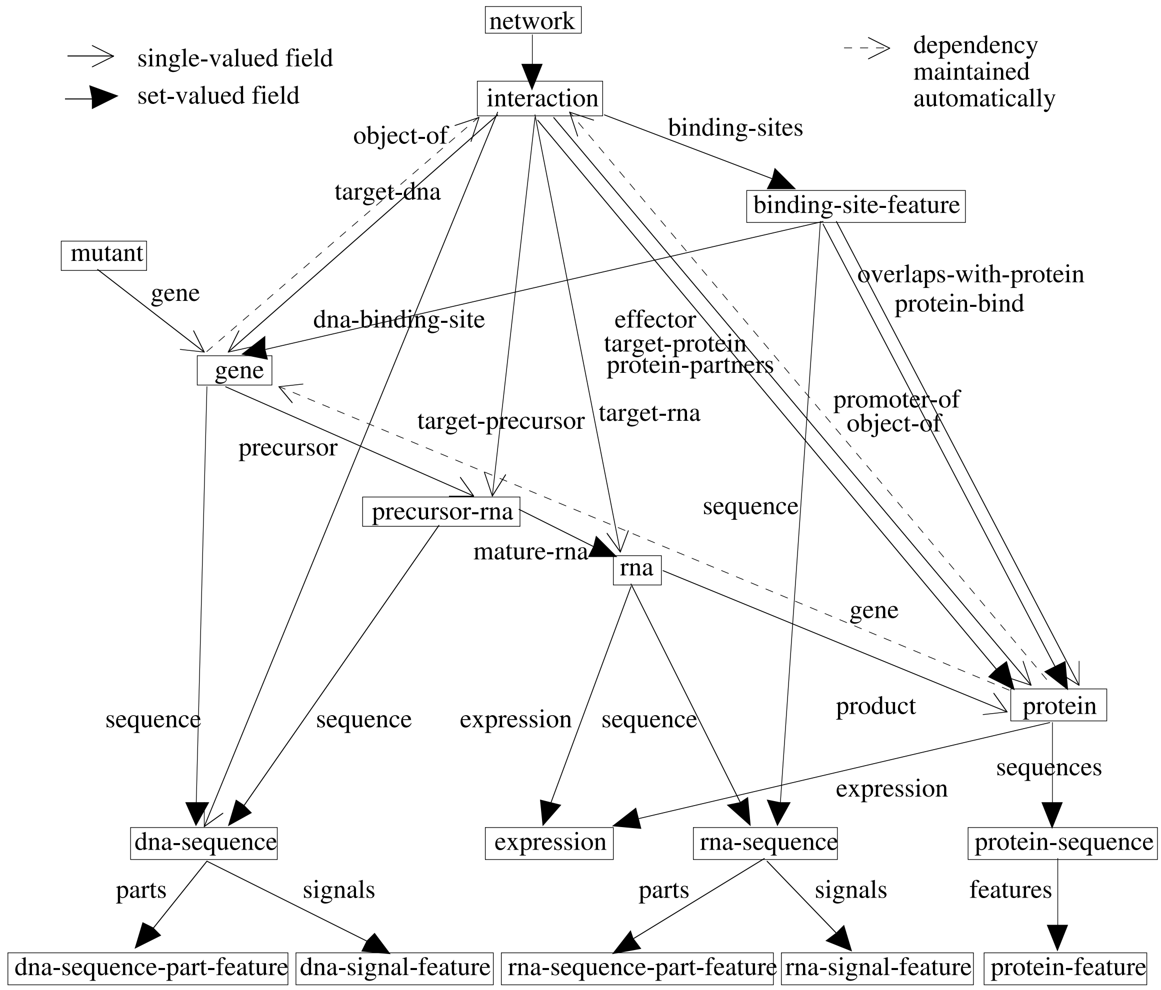

Figure 2. KNIFE concepts and their dependencies. Boxes represent concepts while labelled arrows represent object fields.

signals detected in the sequences (`dna-signal`); these fields contain objects known as `dna-sequence-part-feature` and `dna-signal-feature`. The same applies with slight variations to `rna-sequence` and `protein-sequence` (with no signal/part distinction). They originate from the international sequence databases (EMBL [Rice *et al.* 1993] and SWISSPROT [Bairoch and Apweiler 1996]).

`dna-signal-feature` (2), `rna-signal-feature` (4): describes the various signals (binding sites for regulation proteins, splicing sites) that can be found in the regulatory region of a gene or precursor.

`dna-sequence-part-feature` (2), `rna-sequence-part-feature` (4): describe the functional parts that can be found in a gene. For RNA these are just the parts corresponding to that of DNA.

`protein-feature` (152): the various structural and functional features (zinc-finger, leucine-zipper, homeodomain and helix) of a protein.

Interactions, proteins and genes

The main concept in KNIFE is the `interaction` concept containing currently interactions between proteins and genes (type I). Each interaction connects an `effector` (which is a protein, instance of the `protein` concept) to a `target` (which is a `gene`, instance of the `gene` concept). In addition, the interaction has a `effect` field which contains a string qualifying the interaction effect as `"activation"`, `"repression"` or both. The interactions also mention the following fields:

`binding-sites`: refers to the set of binding sites (concept `binding-site-feature`) involved in the regulation;

`structure-of-target-product`, `structure-of-effector`: refers by a string to the known structural patterns in the products (e.g. zinc-fingers, homeodomains);

`dose-dependence`: indicates if the interaction depends on the concentration of the effector. This is not currently used by the algorithms but will be useful in the future;

`protein-partners`: contains a set of `proteins` which are also involved in the regulation;

other fields contain information from the literature among which the experimental methods used for pointing out the interaction or the regions of the embryo where there is evidence for the interaction [Jacq *et al.* 1997].

Protein-RNA and protein-protein interactions can also be represented. All these interactions are viewed through viewpoints depending on their types (whether they imply protein, RNA or DNA target), the structure of effectors and target product, the effect of the interaction and the classes of the involved products (classified after [Pankratz and Jäckle 1993] for gap genes). The interactions in the base come from GIF-DB [Jacq *et al.* 1997].

The two main elements involved in interactions are proteins and genes. Figure 1 displays three viewpoints on the `protein` concept which has been largely described above. This concept introduces views depending on the initial and final locations of a protein involved, for instance, in the signal transduction pathways; the `biochemical-function` view considers the families of protein biochemical functions.

The `gene` concept refers to the `sequence` of the gene, the `precursor` RNA and the `protein` it codes for through the corresponding fields. It can be viewed under four viewpoints: its `cytological-location`, its phenotypic `class`, its number of precursors and the size of its transcription unit. This is an important issue since the time required by the RNA-polymerase to transcript DNA depends on the size of the unit. This can thus be used in order to impose temporal constraints to the models of embryogenesis (for some homeotic genes, the transcription can last more than one hour).

At start up, KNIFE computes the reverse links from genes and proteins. It is then possible to refer to interactions regulating the gene (incoming) and to interactions in which the gene product is regulator (outgoing).

Genes and proteins originate mainly from FLYBASE [Flybase 1996] and SWISSPROT [Bairoch and Apweiler 1996] and their representation obey the FLYBASE control vocabulary. At the moment, there are 25 genes, 26 proteins and 62 interactions.

Expression patterns

The `expression` concept provides, for a particular development stage and a particular mutation (or wild type individual), the localisation of the expression of a gene, a protein or a particular RNA. Expression patterns are thus identifiable through the following fields:

`developmental-stage` is a string which denotes the stage at which the expression is found;

`gene-name` contains the name of the corresponding `gene` object;

`genetic-context` is a string denoting the kind of mutation (or `"WT"` for wild type).

It provides a set of segments of the embryo in which the product of the gene is found. These patterns are encoded in various ways. The main one is the `egg-length-domain` field which contains the bounds of the intervals of the egg (expressed in percentage of the length of the embryo) where the product is found. For instance, the expression of the `fushi tarazu` gene at the cellular blastoderm stage for a wild individual is made of 7 stripes (shown on Figure 3): 11-16, 20-24, 28-32, 36-40, 44-48, 52-56 and 60-63% of the egg length. The `egg-length-domain` is not the only field containing this information: it is encoded in other common formats (`regions`, `segments` and

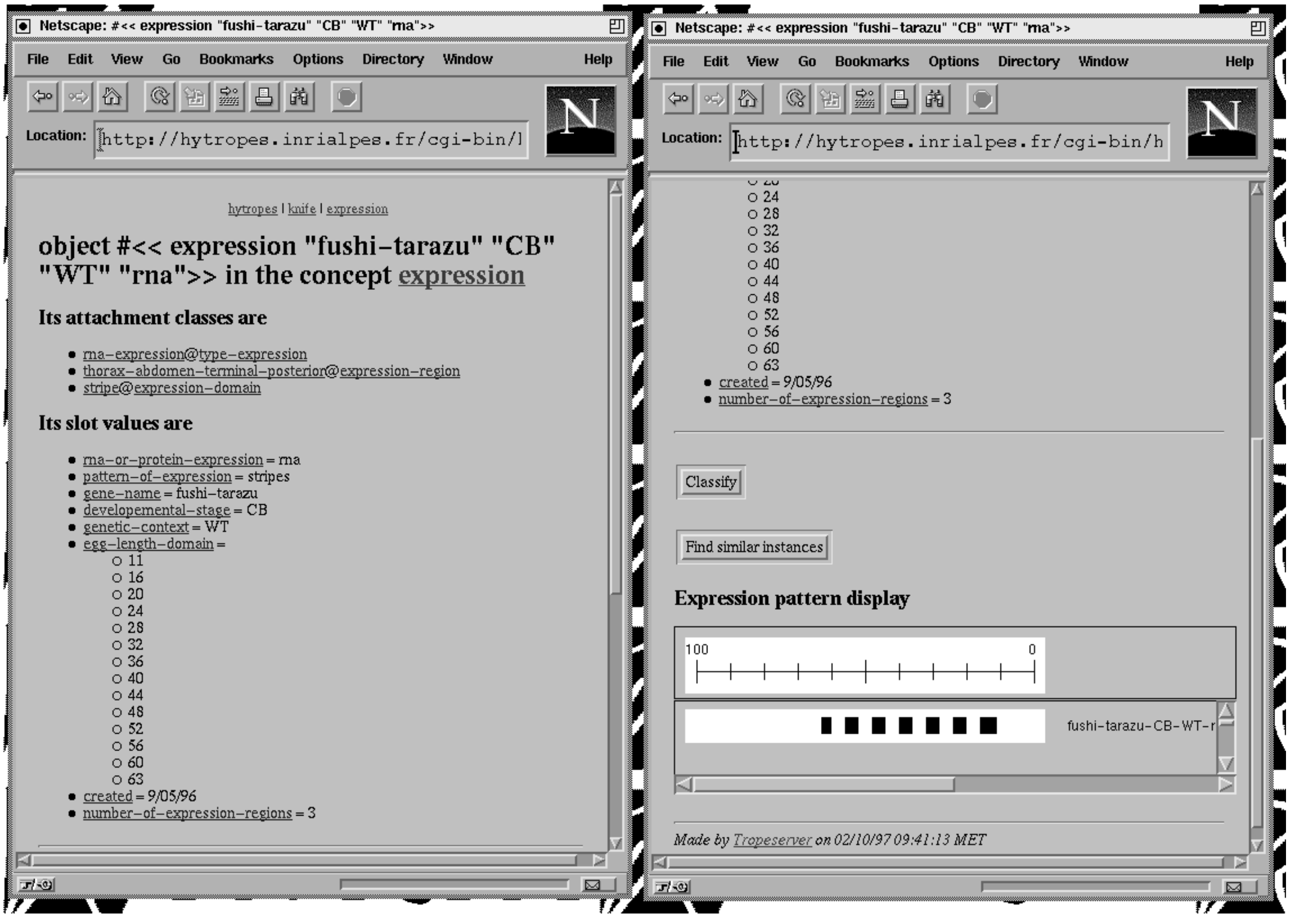

Figure 3. The expression patterns of the `expression` concept are displayed by a Java applet. This allows the users to quickly identify the patterns on the fly egg.

`parasegments`).

This information can be found in the two main views which are offered to the user and which decomposes the expression with regard to the `expression-regions` (the place of expression: `thorax`, `abdomen`, `terminal-anterior`, etc.) and the `expression-domains` (the pattern of expression: `stripe`, `gradient`, etc.).

To date, there are 46 expression patterns coming from the DEXIFLY database [Horn *et al.*, submitted].

Interaction networks

In the KNIFE sense, a network is nothing more than a set of interactions. Its `interactions` field contains the set of `interaction` objects involved in the `network`. It is identified by a `name`. Ideally the networks should be bound to a context expressed through a set of fields:

- `type` is the genetic context of the embryo (wild type or mutant for a particular gene);
- `development-stage` speaks for itself;
- `beginning` and `end` are the beginning and end of egg-length where the network is supposed to occur.

The use of the `exceptions` field, containing a subset of the `interactions`, is described below. There is no network stored as such in the base: building a meaningful network is the goal of manipulating the data.

KNIFE access

An important benefit from the modelling of the knowledge in TROPES is the possibility to automatically obtain a web server, allowing, without any effort from the developer, to manipulate the base. This is the subject of that section while the next one concentrates on the specific algorithms developed for KNIFE and integrated in the web site.

Web availability

TROPES can be transformed into a HTTP server [Euzenat 1996], which means that the knowledge base is automatically turned into a Web site. Each TROPES entity (e.g. object, concept, concept field or class) is provided with an URL and a display function which generates a HTML page (which may contain references to the other entities through their own URL). When such a page is displayed by a HTTP client (or browser), if the user selects

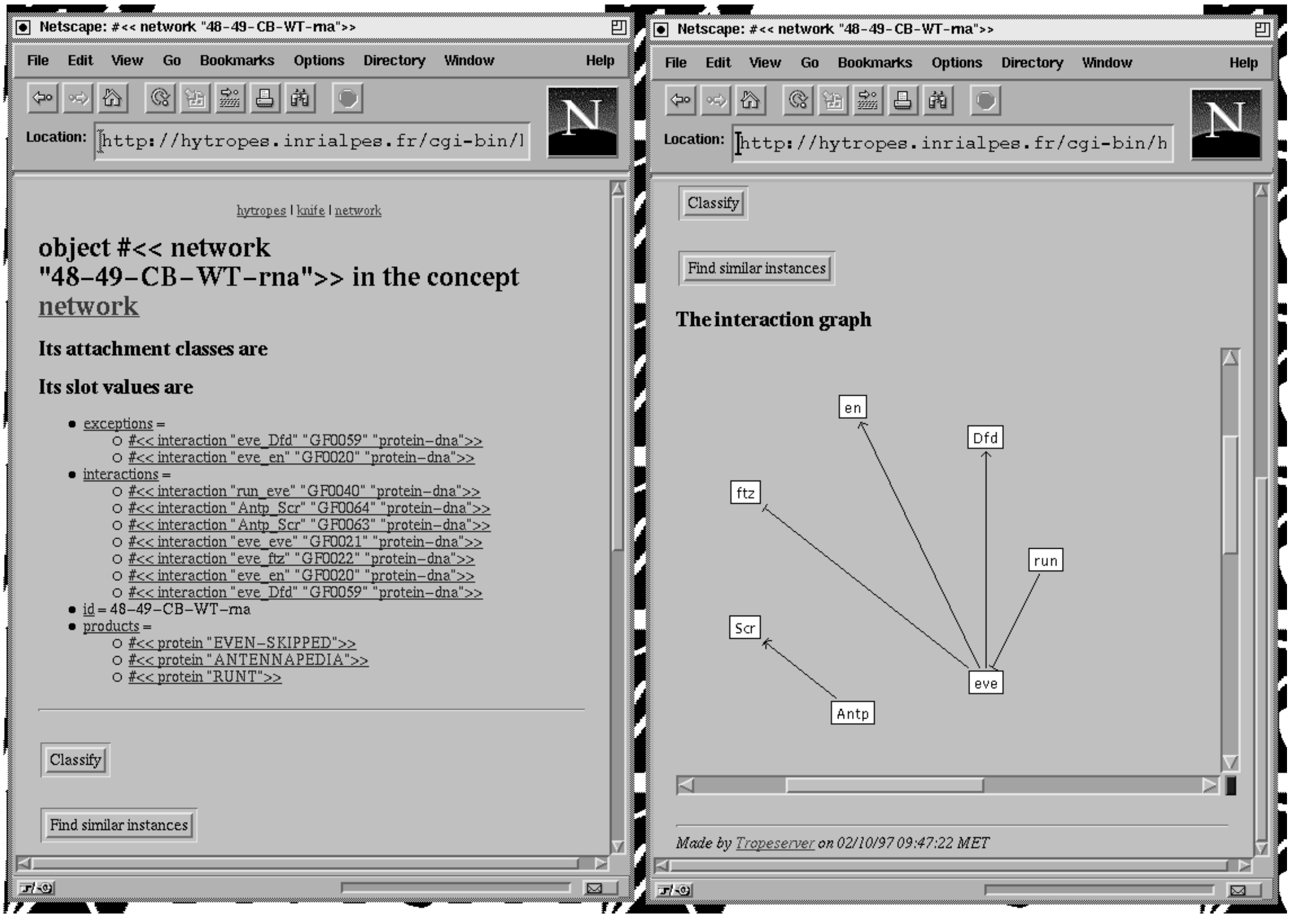

Figure 4. A network. The list of fields is displayed above a graphic representation of the network. Normal arrows (-->) indicate an activating interaction, T-ending arrows (--|) and inhibiting interaction and others (-->|) a twofold interaction. The graph applet can be manipulated by the users so that the display suits their needs.

a particular hypertext link in the page, this will fetch the page corresponding to the embedded URL and this page will be loaded in the client. Each page is generated on demand from the object in the base.

This capability alone has several advantages since it generates a site free of dangling links and allows the remote access to the knowledge base without special training or special computer model. But other, more elaborate, advantages come with some additional work.

First, it is possible to provide annotations to TROPES entities. These annotations can be HTML files which will be incorporated in the usual page generated by TROPES in order to document the displayed object. This provides a first level of enrichment of the Web pages. The included page can be arbitrarily complex (containing URL or images).

Moreover, if the default display pattern of the pages is not suited to the current task, it is possible to redefine it. Figure 3 and Figure 4 show such redefinition with the call, during the generation of the page, to a specialised applet (a program which is run in the HTTP browser) which draws graphic pictures of the considered entity. An important aspect of this is the possibility to automatically generate a call to an applet with the corresponding data. This function also offers the opportunity to generate automatically forms that will send a specific query to an independent database without the burden for the user to construct the query. This is used in KNIFE for providing access to the remote resources familiar to the user community (FLYBASE for genetics, EMBL and SWISSPROT for gene and protein sequences and GIF-DB for interactions).

Queries

From a knowledge base described according to the presentation above, it is possible to apply operations provided by the knowledge representation system. So far, TROPES provides two main operations: classification and filtering. Classification consists, for a particular object, in identifying the classes to which it can be attached; filtering consists, from a particular class, in identifying the objects which could be attached to it. For their respective

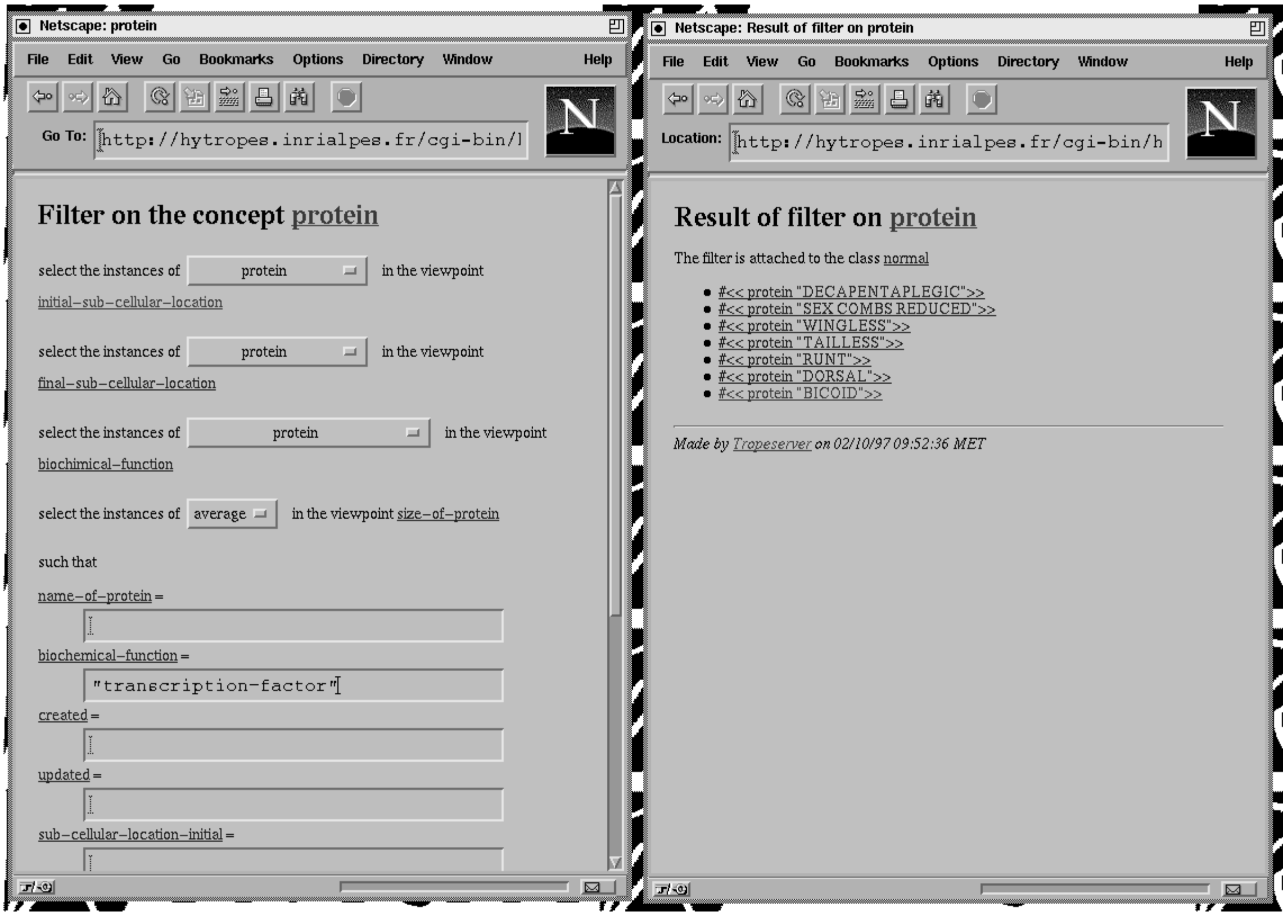

Figure 5. A filtering: the result of filtering the proteins of average size whose biochemical function is a transcription factor is: DECAPENTAPLEGIC, SEX COMBS REDUCED, WINGLESS, TAILLESS, RUNT, DORSAL and BICOID.

purposes, these operations compare the values in the object fields (whatever they are: string, number, objects…) to the constraints attached to the class fields.

TROPES allows to use these reasoning mechanisms in order to issue queries against the base through the Web. For example, filters are used for selecting objects which are attached to certain classes and contain particular field values. These filters are accessible through the HTTP server and allow a more sophisticated search than full text search. For instance, it is possible to ask for all the instances of `protein`, which are classified as `average` in `size` and whose `biochemical-function` is `transcription-factor`. The result is provided in Figure 5.

Network manipulation

The manipulation features described so far are provided in a standard way by TROPES. No particular programming is required, but the knowledge base description. In order to go further some algorithms have been developed specially for the needs of KNIFE. They consist in a graph traversal algorithm and an algorithm for diagnosis of missing interactions.

Network creation and traversing

The KNIFE knowledge base page contains a panel with several operations which apply to the whole base and are specific to the application. First several buttons allow the interactive creation of new networks through the direct selection of the involved interactions or the selection of products which must be source or/and target of the interactions. This allows to create and store sub-networks of particular interest (e.g. the gap-gene network).

KNIFE also allows the enumeration of the paths in a network between a particular product and a particular gene (selected interactively). For instance, there are 51 possible ways for the `BICOID` protein to regulate the `hunchback` gene from the interactions stored in the base (Figure 6). The implemented algorithm is a classical three-passes traversal which proceeds in O(|*N*|) where *N* is the set of nodes in the graph (`protein` objets). It is generally faster than the time required for displaying the result.

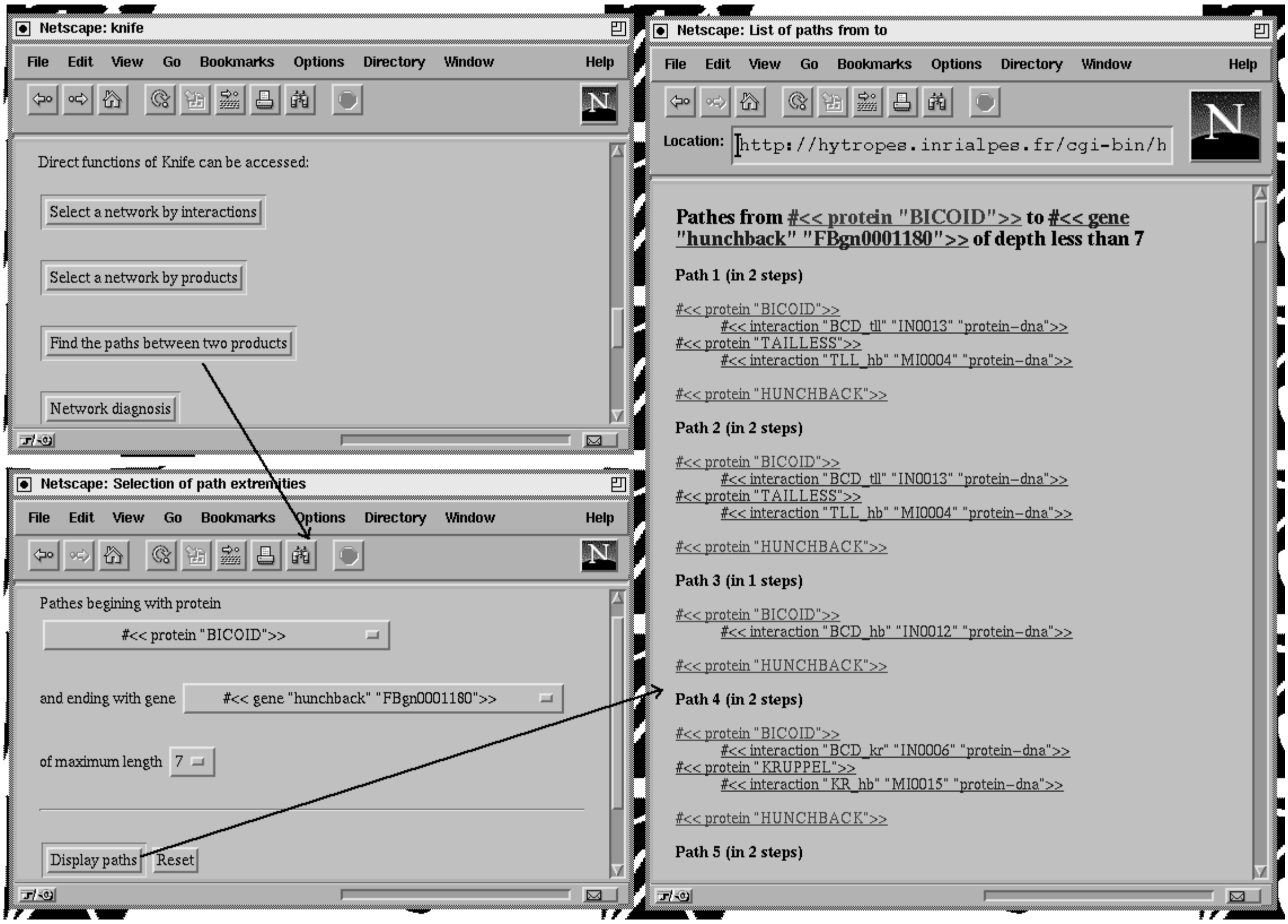

Figure 6. Graph traversing provides all the paths of length less than or equal to 7 between the BICOID protein and the hunchback gene.

Diagnosis of missing interactions

The diagnosis of missing interactions is a new algorithm which could be invaluable for building networks. Its aim is the detection of interactions which should have activated (resp. inhibited) the expression of a gene while this expression is not (resp. is) observed. For that purpose, the user selects the type of fly (e.g. wild type or mutant) and the developmental stage considered (syncitial or cellular blastoderm in the current state of the base). It is also possible to select a particular area (expressed by a segment of the egg) of the fly. The algorithm will then fetch the expression of all the known genes at that stage for that individual and construct a division of the embryo based on all the patterns. For instance, if we consider the genes `Deformed`, `runt`, `fushi tarazu`, `Antennapedia` and `even-skipped`, whose expression patterns (for the wild animal at cellular blastoderm stage) are shown in Figure 7, the result is:

0-9-11-16-17-20-24-25-28-32-33-36-40-41-44-45-48-49-52-56-57-60-63-64-65-68-75-100

Then, the system will build a network for each interval obtained and each network will contain the interactions whose source is a product expressed in the concerned region. For instance, and for the interaction contained in the KNIFE base, the interactions involved between 48% and 49% of the egg length are (--> means activation, --| means repression):

`GF21`: `even-skipped` --> `even-skipped`

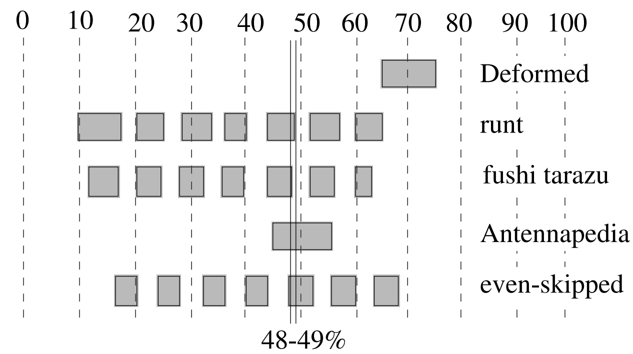

Figure 7. Expression patterns and the considered interval between 48 and 49% of the egg length.

GF22: even-skipped --| fushi tarazu
GF64: Antennapedia --| Sex combs reduced
GF20: even-skipped --> engrailed
GF63: Antennapedia --> Sex combs reduced
GF59: even-skipped --> Deformed
GF40: runt --| even-skipped

Afterward, for each network, the system will point out the products which are expressed although they are inhibited by an active interaction (and activated by no active interaction) or those which are not expressed although they are activated by an active interaction (and inhibited by no active interactions). In the former example, the interactions GF20 and GF59 are in such a case. For instance, GF20 tells that if even-skipped is present, it will activate the expression of engrailed, but engrailed is not present and no other product seems to inhibit it. This means that the base is incomplete on that interaction: either engrailed is expressed (and so its expression pattern is wrong) or something represses its expression.

On another hand, there are no exception on Sex combs reduced which is activated by Antennapedia because there is another repression interaction (from Antennapedia too in that case).

These interactions are put in the exceptions field of the network. This only means that the content of the base does not explain the observations. This is an invitation for the researcher to add new knowledge about interactions in the system or to plan an experiment in order to gather evidence for this lack of knowledge.

If there exists a bound to the number of intervals in a pattern, the complexity of the algorithm is in $O(|N|*|A|^2)$ where A is the set of interactions. There are several ways in which this algorithm could be improved: by targeting the diagnosis on a particular gene or a particular network.

Discussion

The KNIFE knowledge base described above presents several noticeable features:

- the data is type-checked (detection of type-mismatch or misspelling for instance);
- it can be browsed and made available in a client-server fashion (and this is due to the use of a formal representation language);
- it allows the linking of these data with other knowledge sources (other bases, raw data or bibliographic data) while offering its own perspective on the data;
- it provides efficient ways of manipulating the data through filters or classification;
- it allows the usage of specialised algorithms which take their input in the base and deliver their output to the base.

From a biological point of view, it has to be noted that KNIFE is the first knowledge base which is devoted to the description of gene interactions and their networks. Up to now, the amount of biological data present in the base is not yet sufficient to get completely significant results with the algorithms. This conclusion is in fact the result of the use of the algorithms themselves, since they are able to detect inconsistencies in the data. Another problem is that the missing interactions algorithm is too simplistic at the moment since biological interactions cannot be completely described using simple boolean formulas. However, the above algorithm has been designed as a tool for pointing out possible problems and should not be considered as an interaction simulator.

As a matter of fact, the simulation of interactions in the fly embryo is a long-term objective of such a research. No simulation algorithm has been developed so far in the KNIFE knowledge base and this is due to a double requirement: in order to be tractable, the problem should be simplified; in order to be useful, it must not be simplified too much. There are several possible approaches that one could envision in order to tackle the problem.

The first one is boolean simulation which consists, from a network and a set of products (supposed to be present at the beginning), in generating for each possible product the status: present, absent or indeterminate. This type of simulation does not raise any problem but complexity. However, it is not really accurate since it does not account for threshold, efficiency and time, which are important issues in developmental biology.

The second one does take into account the fact that the interactions do not act in a uniform way depending on the concentration of the source in the cell. Moreover, the efficiency and result of interactions is not all-or-nothing but can vary depending on that concentration. Simulating the network on that basis would require the introduction in the base of the knowledge about the thresholds and efficiency as well as the initial quantity of product. The simulation could then be produced in a straightforward manner (provided that we are able to combine the threshold and efficiency of two interactions on the same product) or in a more sophisticated way [Thomas 1991]. However, this solution has the flaw of not considering the time necessary for producing the product and then fails to understand the complete development of interactions.

The deeper possible simulation takes the temporal phenomenon into account. It considers not only the threshold effects and interaction efficiency but also the different delays necessary first to transcribe a gene and to obtain a functional protein and, second, for that protein, to regulate a downstream target. A problematic point with this approach is the present scarcity of biological data. It is perhaps also interesting to note that the problem of the simulation of regulatory networks has some similarities, from a formal point of view, with that of metabolism simulation. Since this latter problem is at the moment the object of intense research work [Karp and Mavrovouniotis 1994; Karp *et al.* 1996a; Karp *et al.* 1996b] it is hoped that future progress in this area will benefit to regulatory network simulation research.

There are three main streams in the future development of the system. The most important one is the feeding of the base with more data. This new data will allow to test the implemented algorithms in context and to evaluate the coverage of the data available. The second direction for improvement is in the user interface: at the moment, a generic interface is proposed by the TROPES system, but it would be better to re-design the actual pages so that they are more adapted to a biologist end-user. This means inclusion of some biological documentation pages and direct links to bibliographic sources for instance. The third important aspect is the development of new algorithms. This covers algorithms for simulating various aspects of genetic regulation and also algorithms for comparing regulation networks between two different development stages or two different organisms.

Acknowledgements

This work is supported by the ACC-SV13 of the French ministry of education and research (MENESR), by CNRS and by INRIA. Many thanks to Ulfar Erlingsson and Mukkai Krishnamoorthy (Rensselaer, Troy) for the graphdraw Java applet used in the KNIFE base (`http://www.cs.rpi.edu/projects/pb/graphdraw`) and Sylvain Chicois for the adaptation of this applet to the needs of KNIFE and a first implementation of the diagnosis algorithm. We also would like to thank Florence Horn and Laurence Röder for all gene spatial expression data, Laurent Fasano and Michel Page for helpful comments. Finally, the authors would like to thank François Rechenmann for continuous support, fruitful discussions and helpful comments.

References

Alexandre, E., Graba, Y., Fasano, L., Gallet, A., Perrin, L., De Zulueta, P., Pradel, J., Kerridge, S., and Jacq, B. 1996. The Drosophila teashirt homeotic protein is a DNA-binding protein and modulo, a HOM-C regulated modifier of variegation, is a likely candidate for being a direct target gene. *Mech. dev.* 59:191-204

Bairoch, A., and Apweiler, R. 1996. The SWISS-PROT protein sequence data bank and its new supplement TREMBL. *Nucleic Acids Res.* 24:21-25

Euzenat, J. 1996. Knowledge bases as Web page backbones. In proceedings 5th WWW workshop on "artificial intelligence-based tools to help W3 users", Paris (FR) `http://www.inrialpes.fr/sherpa/papers/euzenat96a.html`

Fasano, L., Röder, L., Coré, N., Alexandre, E., Vola, C., Jacq, B., and Kerridge, S. 1991. The gene teashirt is required for the development of Drosophila embryonic trunk segments and encodes a protein with widely spaced zinc finger motifs. *Cell* 64:63-79

The Flybase consortium 1996. FlyBase: a Drosophila database. *Nucleic acid research* 24:53-56

Gaul, U., and Jäckle, H. 1990. Role of gap genes in early development. *Adv. Genet.* 27:239-275

Hoogland, C., and Biémont, C. 1997. Drosoposon: a knowledge base on chromosomal localization of transposable element insertions in Drosophila. *CABios* 13(1):61-68

Jacq, B., Horn, F., Janody, F., Gompel, N., Serralbo, O., Mohr, E., Leroy, C., Bellon, B., Fasano, L., Laurenti, P., and Röder, L. 1997. GIF-DB: a WWW database on gene interactions on gene interactions involved in Drosophila melanogaster development. *Nucleic Acids Res.* 25:67-72

Karp, P., and Mavrovouniotis, M. 1994. Representing, analyzing, and synthetizing biochemical pathways. *IEEE Expert* 9(2):11-22 `http://www.ai.sri.com/cgi-bin/pubs/papers/Karp94-11:Representing/document.ps`

Karp, P., Riley, M., Paley, S., and Pelligrini-Toole, A. 1996a. *Nucleic Acids Res.* 24:32-40

Karp, P., Ouzounis, C., and Paley, S. 1996b. HinCyc: a knowledge base of the complete genome and metabolic pathways of *H. influenzae*. In proceedings 4th ISMB, St Louis (MO US) `http://www.ai.sri.com/cgi-bin/pubs/papers/Karp96:HinCyc/document.ps`

Lewis, E. 1978. A gene complex controlling segmentation in Drosophila. *Nature*, 276:565-570

Mariño, O., Rechenmann, F., and Uvietta, P. 1990. Multiple perspectives and classification mechanism in Object-oriented Representation. In proceedings 9th ECAI, Stockholm (SE), pp425-430

OMIM Team 1996. Online Mendelian Inheritance in Man (OMIM), a catalog of human genes and genetic disorders. Johns Hopkins University, Baltimore (MD US) and National Center for Biotechnology Information, Bedestha (MD US) `http://www3.ncbi.nlm.nih.gov/omim/`

Nüsslein-Volhard, C., and Wieschaus, E. 1980. Mutations affecting segment number and polarity in Drosophila. *Nature* 287:795-801

Pankratz, M., and Jäckle, H. 1993. Blatoderm segmentation. In Bate, M., Martinez-Arias, A., eds., The development of *Drosophila melanogaster*, pp467-516 Cold Spring Harbor Laboratory press

Perrière, G., Dorkeld, F., Rechenmann, F., and Gautier, C. 1993. Object-oriented knowledge bases for the analysis of prokaryotic and eukaryotic genomes. In proceedings 1st ISMB, Bethesda (MD US), pp319-327

Rice, C., Fuchs, R., Higgins, D., Stoehr, P., and Cameron, G. 1993. The EMBL Nucleotide sequence Database. *Nucleic Acids Res.* 21:2967-2971

Röder, L., Vola, C., and Kerridge, S. 1992. The role of the teashirt gene in trunk segmental identity in Drosophila development. *Development* 115:1017-1033

SHERPA project 1995. TROPES 1.0 reference manual, Internal report, INRIA Rhône-Alpes, Grenoble (FR), 85p. `ftp://ftp.inrialpes.fr/pub/sherpa/rapports/tropes-manual.ps.gz`

Thomas, R. 1991. Regulatory networks seen as asynchronous automata: a logical description. *J. theor. Biol.* 153:1-23