MotifML: A Novel Ontology-based Language for Protein-Binding Motifs: Implications in Dissecting the Genetic Regulatory Circuitry

Running title: MotifML: A Novel Language for DNA Motif Profiles

Tianhua Niu*, Bioinformatics Group, Division of Preventive Medicine, Department of Medicine, Brigham & Women's Hospital, Harvard Medical School, Boston MA 02215 USA

Kenneth Baclawski, College of Computer Science, Northeastern University, Boston, MA 02115 USA

Sui Huang, Department of Surgery, Children's Hospital, Harvard Medical School, Boston, MA 02115 USA

Jerzy Letkowski, Western New England College, Springfield, MA 01119 USA

*Corresponding author:

Tianhua Niu

900 Commonwealth Avenue East

Boston MA 02215-1213 USA

Tel: (617) 432-2985

Fax: (617) 432-2956

E-mail: tniu@rics.bwh.harvard.edu

Keywords: ontology, cis-regulatory element, computational biology, DNA motifs

Abstract

DNA micro-array technology has tremendously accelerated the pace of identification of critical DNA sequence motifs in both prokaryotic and eukaryotic organisms. Sophisticated statistical algorithms, such as the Hidden Markov Model, have been applied to the problem of discovering motifs. Such algorithms have been implemented in a number of bioinformatics software tools including Gibbs Motif Sampler, AlignACE, BioProspector, and CONSENSUS. These computer programs have been successfully used to identify co-regulated genes in bacteria and yeast. However, these bioinformatics tools differ markedly in their output formats as well as differing in the semantics of their results. As a result there is no systematic framework that facilitates efficient data exchange, querying, consistency checking and merging of DNA motif profiles, especially when the profiles originate from several sources. We propose a novel language that can address these critical problems such that motif profiles from different species identified using different software tools and algorithms can be integrated and compared with each other as well as with data formatted in other bioinformatics languages. The proposed language also addresses the problem of representing interactions among regulatory regions so that they can be simulated and visualized. The adoption of such a language will expedite and formalize the process of uncovering and annotating co-regulated genes at different developmental stages and can bring new insights into our current understanding of biopathways.

1 Introduction

Because experimental approaches are increasingly capable of providing reliable data pertaining to gene regulation, theoretical models are becoming important in the understanding and manipulation of such processes. Delineation of the transcriptional networks controlled by transcription factors is a major goal of functional genomics (Wen et al., 1998, Tavazoie et al., 1999).

One may consider the arrangement of transcription factor (TF) binding sites to be "sentences of words" encoded in the DNA for controlling such important biological functions as cell growth, proliferation and differentiation. The core "words" of these "sentences" are the DNA binding motifs for their respective TFs. Expression data generated from microarray studies, in conjunction with a comprehensive collection of genes from genomic sequence resources, promises to accelerate the discovery of new motifs.

1.1 Promotors and Transactivation

Theoretically, in a simplified way one can describe the relationship between promoter region and transactivation of a given gene as follows:

The promoter activity P (measured as transcription rate) of a given gene is a function f of the input configuration vector $\mathbf{C}$ which describes occupancy of the regulatory sites:

$$P = f(\mathbf{C})$$

$$\mathbf{C} = (M_1, M_2, \ldots, M_n), \quad \text{where } n \text{ is the number of motifs in the promotor region}$$

$$M_i = \{p_1, p_2, \ldots, p_k\} \quad \text{is the occupancy state of motif } i \text{ with proteins } p_1, p_2, \ldots$$

$$M_i = \{\} \qquad \text{means that the site is unoccupied.}$$

Thus, a configuration could be: $\mathbf{C} = (\{aa\}, \{b\})$, indicating that site M_1 is bound by a dimer of protein a, and site M_2 is bound by protein b. The function f fully characterizes the "promoter logics" by mapping each input configuration $\mathbf{C}$ into a value of P.

In reality, f is not known. Information about the functional relationship between the regulatory regions is usually very sparse, since the elucidation of such relationships requires tedious experimentation. One well characterized example is the regulation of *Endo16* gene in sea urchin (Yuh et al., 1998).

A complete description would require the molecular testing of all configurations $\mathbf{C}$ by experimental controlling the presence or absence of the individual transcription factors and associated proteins. This can in principle be done by genetic manipulation of the cell or the organism, e.g., by gene knockout mutants or overexpressing transgenic mutants. Clearly, a systematic analysis with such experiments is currently beyond the scope of available techniques.

It is simpler, to characterize promoter regions at the DNA level. Although such approaches are crude and rely on artificial situations, some valuable information on the functional relationships can be obtained. Hereby, motifs an artificial promoter/reporter gene construct are mutated within instead of controlling the transcription factor availability. This

manipulation at the DNA level is routinely used to functionally characterize regulatory regions. Thus, for a long time to go, any annotation of motif interactions will have to capture the partial knowledge provided by this type of promoter analysis and reflect the experimental data given the technical limitations, rather represent the complete "true" relationships as discussed above theoretically.

The type of information that is available from the current type of promoter analysis essentially reduces the scope of functional interactions that can represented in MotifML to the following:

1. *Interaction at DNA level*

   (a) *Deletion mutants*

   If one motif (M_1) is deleted, how is P affected? This is specified by a numerical value giving the increase / decrease factor. If the experiment shows that P is reduced to close to zero, then M_1 can be labeled as a motif that is *necessary* for transactivation.

   (b) *Independent reporter gene activation*

   Has a short DNA stretch containing just the motif M_1, or oligomers thereof, been shown to drive transactivation of a reporter gene, constitutively (basal value of P) or conditionally (inducible value of P)? If yes, then the said motif can be labeled as *sufficient* to drive basal, or conditional expression, respectively.

(c) *Cooperativity*

Is the value of $P(M_1, M_2)$ significantly larger than $P(M_1) + P(M_2)$? If experiments exist that show this, then the motifs M_1, and M_2 can be assigned the label "cooperative interaction".

(d) *Inhibition*

The presence of a motif, M_k at a certain position relative to M_1 could be inhibitory for the function of M_1. This would experimentally be evidenced in deletion mutants in which the deletion of M_k would increase the value of P. One example of such "silencers" are the apparent NRDs (negative regulatory domains) of interferon genes. However, in eucaryotes, most inhibition is actually mediated by the binding of alternative (non-activating) factors to an otherwise activating site, for instance, IRF-2 binding to ISRE instead of IRF-1. Thus, the functional relationship is determined at the protein level.

2. *Interaction at protein level*

As mentioned above, experimental data on the interaction of transcription factors and their functional consequences are sparse. Still, some type of information is available, allowing annotation of motifs for the following categories of interaction:

(a) *Identity of binding proteins*

A motif can be annotated with the set of proteins that have been shown to bind

to the motif in gel shift experiments.

(b) *Physical protein-protein interaction*

In some cases two adjacent motifs M_1 and M_2 exhibit a functional interaction because the proteins physically interact. For the annotation of such protein-protein an experimental demonstration of the protein(-DNA)-complex is needed. Protein-protein interactions can occur in the absence or (only) in the presence of DNA. The latter case typically is associated with cooperativity (see DNA level interaction). For the protein-protein interaction, one might envisage an interface with protein-interaction databases.

The following are the general principles that were considered in our annotation efforts to bridge the gap between the motifs of individual genes and *higher level biological function*:

(1) The identity of the transcription factor that binds to the motif.

(2) The existence of alternative binding proteins that can inhibit/attenuate the transcription.

(3) The conformation of the transcription factor that binds to the motif, monomer, homodimer or heterodimer.

(4) Whether there is positive feedback or pleiotropy. This principle is especially important. It provides the basic information with which in the future genetic circuits and networks can be built.

(5) The biological significance.

1.2 The Goals of MotifML

Although the binding specificity of TF is often expressed by means of a consensus sequence, a position weight matrix (PWM) is usually more appropriate. The PWM assigns a weight to each possible nucleotide at each position within the site, reflecting the frequency with which the given nucleotide occurs at the given position. The score of a particular site is obtained by summing the corresponding weights. This captures more information than an oversimplified consensus, and has a sound foundation in both statistics (representing likelihood ratios) and thermodynamics (representing binding energies).

While there are experimental methods to identify regulator elements, they are expensive, time consuming and technically difficult. Thus, computational methods exploiting the vast databases of DNA sequences generated from the genome sequencing projects are an attractive alternative. Recently statistical models have been developed to quantitatively model the clustering of sites often seen in regulatory regions. The most successful statistical model has been the Hidden Markov Model (HMM), in which the hidden state signals whether the current nucleotide is or is not within a regulatory region. Within regulatory regions the model expects shorter average spacing between sites, so that clusters of sites are more likely to be modeled as a regulatory region. A number of computer programs are available based on the HMM, and the currently most popular ones include AlignACE, Gibbs Motif Sampler, BioProspector and CONSENSUS.

The various motif-searching tools described above differ from one another not only in their algorithms for motif identification but also in their output formats. This makes it difficult to compare and combine their respective results. We therefore are proposing a general knowledge representation language *MotifML* that can be the basis for an integrated framework. The MotifML language can not only encode every type of motif, but it can also compare and combine motifs. We present the MotifML language as well as a prototype graphic user interface (GUI) that can be used to facilitate reasoning about the genetic regulatory circuitry and that is amenable to efficient computation.

The specific goals of MotifML are:

- To allow the full specification of all experimental information known about motifs;
- To provide an extensible framework for this annotation;
- To provide a common vehicle for exchanging the motif information; and
- To allow reasoning (i.e., inference), querying, consistency checking and merging of experimental information about motifs from diverse sources.

Having represented information, it is important for it to be easily retrievable. This is most commonly done via query processing. Because of the complex nature of biological information, it is necessary to allow for the system to infer information that is not explicitly stored. The most common form of inference is specialization/generalization (also called

subsumption), but part-whole relationships and other relationships are also important for inference. Inference is also called *reasoning*.

Although support for reasoning is essential, information must have high quality for inference to be dependable. In particular, the information must be *consistent*. If information is inconsistent, then anything may be inferred, and so no conclusions (including the results of queries) can be trusted.

Another important aspect of reasoning for biological information is that much of it is not completely certain. Issues such as trust and belief are also important. Reasoning using such information must take into consideration the degree of certainty of the statements and must propagate this certainty to conclusions based on the statements. When there are conflicting statements, they must be reconciled by adjusting their certainty levels. Trust and belief can also be propagated in this way.

Merging information from diverse sources is especially difficult because of the likelihood of inconsistencies. Introducing measures of certainty, trust and belief, is a mechanism for reconciling such conflicts.

1.3 Ontologies

In order to store biological knowledge it is necessary to represent it in a computer-compatible fashion. This is a special case of data storage in general. The structure of the data in a database is defined by its *schema*. The schema for a knowledge-based system is called its *ontology*. An ontology is an explicit, formal, machine-readable semantic model. We were

responsible for some of the earliest work on the specification of biological ontologies and the automated annotation of biological documents, as discussed in (Baclawski et al., 1993a, Baclawski et al., 1993b, Hafner et al., 1994). We were also among the first to address the problem of visualizing knowledge representations both in general and for biological knowledge in particular (Baclawski et al., 2000).

To specify an ontology one must have an ontology specification language. There are currently many ontology specification languages. We discuss and compare these languages in section 2 below. The increasing interest in ontologies is driven by many forces, but one that is rapidly gaining momentum is the emergence of web-enabled agents (McGuinness, 2001). These agents can reason about and dynamically integrate the appropriate knowledge and services at run-time based on formal ontologies. Ontologies are also the basis for the Semantic Web which has been proposed in (Berners-Lee et al., 2001). We have been developing tools for constructing and checking ontologies in general, but especially the ontologies used by the Semantic Web (Baclawski et al., 2001a, Kogut et al., 2002, Kokar et al., 2001).

2 Methods

To achieve the goals of MotifML, it is necessary to design a language that can effectively specify experimental information about motifs. In this section this language is designed. The syntax and semantics are available online at (MotifML, 2002). In section 3 it is shown that the goals have been achieved by mapping the output of a major motif-related system

to the MotifML language.

There are many possibilities for designing a language. However, the requirement that the language be extensible and a common vehicle for exchange of motif information limits the choices to languages supported by the World Wide Web (WWW). The WWW has become the de facto common platform for the exchange of any kind of data. Nevertheless, there are still many choices even within the languages supported by the WWW. The eXensible Markup Language (XML) is a "language of languages" that permits the definition of markup languages. Each XML language is defined by a Document Type Definition (DTD) that specifies what kind of elements can appear in a conforming document, what attributes each element may have and how elements can be contained within each other. XML documents can be deep and semantically rich. A single XML document can contain numerical data, text and sequence information, all of which are tagged so that the meaning of every item of information in the document is unambiguously specified. XML is a recommended standard of the WWW for data interchange (W3C, 2001a). Many tools are being developed for XML, and XML parsers are readily available. Furthermore, a powerful XML transformation language called XSLT (W3C, 2001c), has been developed. XSLT that can be used to transform XML documents from one DTD to another DTD, as well as to query XML documents for specific information. Many XSLT tools are available both commercially and in the public domain.

Many XML languages have now been developed for biomedical information of various kinds, such as BioML, CellML, Bioinformatic Sequence Markup Language (BSML), System

Biology Markup Language (SBML), DNA Markup Language (DNAML), and so on. These languages offer a solution for storage and interchange of data in a uniform and automatically parseable format (Barillot and Achard, 2000). However, in spite of the many advantages of XML over languages and formats that preceded it, there are still many limitations:

(1) It is very difficult to extend an XML DTD. Tools that recognize and process a particular XML DTD can only recognize extensions that are already provided by the original design of the language. Unanticipated extensions cannot be processed.

(2) Linking information in different documents is difficult, even when documents use the same DTD. If the documents use different DTDs, such links are even more difficult and may be impossible.

(3) Merging information from different markup languages is difficult.

(4) XML does not support general relationships such as many-to-many relationships.

Recognizing these limitations, the WWW has been engaging in the development of new XML-based languages. The most recent proposal is called the Web Ontology Language (OWL) (Smith et al., 2002). OWL is an emerging standard for ontologies and knowledge representations, based on the Resource Description Framework (RDF) (Lassila and Swick, 1999) and the DARPA Agent Markup Language (DAML) which is the immediate predecessor of OWL. Many tools have been and are being developed for OWL. The following are some of the features supported by OWL:

(1) OWL is declarative and formally defined.

(2) It is easy to create OWL documents using readily available tools.

(3) OWL is compatible with commonly used network formats and protocols, especially with the formats and protocols of the WWW.

(4) OWL is interoperable with existing biomedical languages and formats, especially those based on XML.

(5) It is easy to extend a OWL-based language.

(6) One can use OWL to link items anywhere on the WWW, not just within the same document and same DTD as in XML.

(7) There is a mechanism for merging information from different sources.

(8) RDF and OWL fully support specialization/generalization hierarchies.

(9) RDF and OWL support arbitrary relationships, including many-to-many relationships.

OWL was developed for ordinary logical assertions and facts. For numerical and data formatting constraints, it uses XML Schema Part 2 (W3C, 2001b).

In order to interoperate with the many algorithms that can be used to find motifs, the various formats must be transformed into a common format. This is done using transformations tools. When the source format is in a text format, such as the output formats of

AlignACE, Gibbs Sampler, CONSENSUS and BioProspector, we use Perl scripts to transform the output to MotifML. Transforming MotifML to other formats is accomplished by using XSLT (W3C, 2001c).

3 Results

The top-level taxonomy of MotifML is shown in Figure 1. MotifML imports other, more generic, ontologies for biology that are not shown in this Figure. Some concepts, such as *Gene* are also defined in one of the imported ontologies, and this concept is equivalent to the one in the imported ontology.

[Approximate location of Figure 1.]

The ontology also contains many attributes and relationships, some of which are shown in Figure 2. For simplicity, inherited attributes are not shown. The detailed constraints on the various relationships were also not shown.

[Approximate location of Figure 2.]

3.1 Examples of Motifs

An example of part of a typical motif (found by BioProspector) is shown in Figure 3. This particular motif is a regulatory motif for a gene coding for the HSP70 heat shock protein. The rows in the top table represent probability distributions on the DNA bases, each row being one position in the motif. Later rows show particular examples of regulatory regions of specific genes. The corresponding MotifML representation is shown in Figure 4. Although

this particular motif is ungapped, the MotifML representation supports gapped motifs.

[Approximate location of Figure 3.]

[Approximate location of Figure 4.]

An example of the graphic user interface for MotifML is shown in Figure 5.

[Approximate location of Figure 5.]

3.2 Consistency Checking and Uncertainty Propagation

As discussed in the goals for MotifML in Section 1.2, it is important for information to be consistent. Checking consistency is, in general, very difficult. Computationally, consistency checking is *undecidable*, meaning that there is no algorithm that can check consistency in a finite amount of time. Nevertheless, there are techniques that can effectively check consistency in all cases. The ConsVISor consistency checking tool (Kokar et al., 2001) is an example of such a tool that was designed for checking consistency of information expressed using a OWL ontology.

Specifying uncertainties and the propagation of uncertainty in knowledge-based systems has also been addressed by the authors. This is done by first expressing the information without uncertainty. This is known as the *crisp* formulation. In other words, one makes statements as if they were definitely and unambiguously known to be true. One then chooses an ontology for expressing uncertainty. There are several frameworks for uncertainty from which one can choose. For example, classical probabilistic uncertainty has excellent theoretical and empirical foundations. However, it does not handle complex situations very well.

The simplifications (such as independence assumptions and prior distributions) that are typically assumed, negate much of the theoretical justification for this framework. Another example is fuzzy reasoning. This approach scales much better to complex situations, but it is not as well founded as probabilistic uncertainty. There are still other frameworks such as Dempster-Shafer belief theory.

MotifML is currently configured to use probabilistic uncertainty. The ontology for any form of uncertainty requires that one define the propagation of uncertainty for the following operations:

1. Propagation via inference (if-then) rules.

2. Boolean operations (AND, OR, NOT).

3. Arithmetic operators.

4. Type-specific operations that are not easily expressed using arithmetic operations.

Propagation via inference rules is most commonly formalized using conditional events as in (Goodman et al., 1997). The other operators mentioned above have been formalized for fuzzy logic in (Li et al., 1999).

One must generally make assumptions about the dependencies among events to make the fusion computations possible. Common assumptions include:

1. Independence assumption. This is often a reasonable estimate even when independence is not known.

2. Conservative assumption (lower bound).

3. Liberal assumption (upper bound).

Independence is assumed unless there is an explicitly stated dependence (inhibitory or co-operative) as discussed in Section 1.1.

Some examples of inconsistencies that can be discovered and resolved include:

1. Misspellings (typos). One might, for example, incorrectly enter the name of a transcription factor. These are difficult to distinguish from similarly named transcription factors that are actually different. As a result they are handled by printing warnings which the experimenter must check manually.

2. Motifs generated from the same data set by different algorithms. The algorithms use very different assumptions, so they can produce markedly different results. These and the next two are resolved by combining uncertainties as discussed above.

3. Motifs generated from different data sets by the same algorithm.

4. Motifs generated by different runs on the same data set using the same algorithm. These are not usually so marked, and they are already handled by probability estimates produced by the algorithm.

5. Combinations of several of the above sources of inconsistency.

4 Discussion

Precise analysis of the genetic network, gene function and transcription regulation requires accurate prediction of TF binding motifs. we have presented a motif representation language for the results of four different programs that have enjoyed widespread use in the computational biology field: AlignACE, Gibbs motif sampler, BioProspector, and CONSENSUS. It is currently difficult to use more than one of these programs effectively because the algorithms used by popular programs all use incompatible formats and semantics.

MotifML offers a solution to store and describe data in a uniform and automatically parseable format. Although several other XML-based bioinformatics markup languages are currently available, such as the CellML, Bioinformatic Sequence Markup Language (BSML), and System Biology Markup Language (SBML), none of them can handle the heterogeneity present in the data of motif discovery. The key feature of MotifML is in categorizing and defining the motif data found by different motif searching algorithms in a context-insensitive manner. Furthermore, MotifML can support probabilistic inference, thus enhancing its value in "knowledge mining" information dealing with genetic regulatory circuitry.

[Approximate location of Figure 6.]

In Figure 6 we give a diagrammatic illustration of this process. Each of the three major columns represents either a step in a time course in the evolution of a cell type or (more commonly) cells from several individuals. The two minor columns in each major column represent multiple cell lines. The next two rows represent mRNA extraction and reverse

transcription to cDNA. The cDNA is then hybridized to produce micro-array data, which is clustered to obtain correlations. These correlations are then processed to find regions that are coregulated by the same transcription factor, i.e., the motifs. The motifs may be found using one of the many motif discovery algorithms such as Gibbs Motif Sampler, AlignACE, BioProspector, and CONSENSUS discussed above. Each of these produces an output file. The output files are then converted to MotifML and merged into a single knowledge representation. The topology of genetic regulation may then be viewed and navigated using the MotifML tool. MotifML furnishes a uniform language for relating the knowledge discovered in these parallel strands. When combined with our PathML language for biomedical pathways (Baclawski et al., 2001b), MotifML supports basic research for formulating and testing scientific hypotheses concerning genetic pathways. The ultimate goal is to find methods for treating and curing diseases. MotifML and PathML are intended to be an infrastructure for biomedical hypothesis generation and testing.

One of the major goals of our efforts is the modeling transcriptional regulatory networks in eukaryotes. Such models would allow for online dynamic simulations of transcriptional regulatory networks based on MotifML knowledge representations. We also plan to extend MotifML to incorporate richer dynamic models and forms of uncertainty for transcriptional regulatory networks as well as to support semantic links with other biological markup languages. Finally, we will make our tools freely available to the community and to support distributed annotation over the Web, as in Lincoln-Stein and Open-Bio.

Bioinformatics is the confluence of both computational and biological disciplines. This prototype was designed to allow a multidisciplinary approach to the use of bioinformatics in order to engage in scientific inquiry in the discovery of information related to protein-binding motifs. Providing the option to select both file and format enables the user to direct access control. Providing full customizability of the code enables the programmer to direct processing control. At the final stage, providing an interactive visual display of biological information enables the scientist to direct decisional control. Allowing each type of user the ability to select and customize the process improves the chances of usefulness of the product as well as potentially decreasing the time and cost of the investigation (Schneiderman, 1998).

5 Future Directions

Acknowledgements

We would like to thank Prof. J. S. Liu (Department of Statistics, Harvard University) for providing the Linux platform in implementing the various motif searching algorithms.

References

[Arnold et al., 1994] Arnold, D., Feng, L., Kim, J., and Heintz, N. (1994). A strategy for the analysis of gene expression during neural development. *Proc. Natl. Acad. Sci. U.S.A.*, 91(21):9970–9974.

[Baclawski et al., 2000] Baclawski, K., Cigna, J., Kokar, M., Mager, P., and Indurkhya, B. (2000). Knowledge representation and indexing using the Unified Medical Language System. In *Pacific Symposium on Biocomputing*, volume 5, pages 490–501.

[Baclawski et al., 1993a] Baclawski, K., Futrelle, R., Fridman, N., and Pescitelli, M. (1993a). Database techniques for biological materials & methods. In *First Int. Conf. Intell. Sys. Molecular Biology*, pages 21–28.

[Baclawski et al., 1993b] Baclawski, K., Futrelle, R., Hafner, C., Pescitelli, M., Fridman, N., Li, B., and Zou, C. (1993b). Data/knowledge bases for biological papers and techniques. In *Proc. Sympos. Adv. Data Management for the Scientist and Engineer*, pages 23–28.

[Baclawski et al., 2001a] Baclawski, K., Kokar, M., Kogut, P., Hart, L., Smith, J., Holmes, W., Letkowski, J., and Aronson, M. (2001a). Extending UML to support ontology engineering for the Semantic Web. In Gogolla, M. and Kobryn, C., editors, *Fourth International Conference on the Unified Modeling Language*, volume 2185, pages 342–360. Springer-Verlag, Berlin.

[Baclawski et al., 2001b] Baclawski, K., Niu, T., and Neumann, E. (2001b). Using DAML format for representing complex gene networks: implications in novel drug discovery. *Amer. J. Human Genetics*, 69 (Suppl)(4):457. Presented at the 51^{st} Annual Meeting of the Amer. Soc. Human Genetics in San Diego, CA on October 12–16, 2001.

[Barillot and Achard, 2000] Barillot, E. and Achard, F. (2000). XML: a lingua franca for science? *Trends in Biotechnology*, 18:331–333.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*.

[Goodman et al., 1997] Goodman, I., Mahler, R., and Nguyen, H. (1997). *Mathematics of Data Fusion*. Kluwer Academic Publishers, Dordrecht, Netherlands.

[Hafner et al., 1994] Hafner, C., Baclawski, K., Futrelle, R., Fridman, N., and Sampath, S. (1994). Creating a knowledge base of biological research papers. In *Proc. Second Int. Conf. Intell. Sys. Molecular Biology*, pages 147–155.

[Kogut et al., 2002] Kogut, P., Cranefield, S., Hart, L., Dutra, M., Baclawski, K., Kokar, M., and Smith, J. (2002). UML for ontology development. *Knowledge Eng. Rev.*, 17(1):61–64.

[Kokar et al., 2001] Kokar, M., Letkowski, J., Baclawski, K., and Smith, J. (2001). The ConsVISor consistency checking tool. `www.vistology.com/consvisor/`.

[Lassila and Swick, 1999] Lassila, O. and Swick, R. (1999). Resource description framework (RDF) model and syntax specification. `www.w3.org/TR/REC-rdf-syntax`.

[Li et al., 1999] Li, J., Kokar, M., and Weyman, J. (1999). Incorporating uncertainty into the formal development of the fusion operator. *Proc. Second Int. Conf. Information Fusion*, pages 125–132.

[McGuinness, 2001] McGuinness, D. (2001). Ontologies and online commerce. *IEEE Intelligent Systems*, 16(1):8–14.

[MotifML, 2002] MotifML (2002). Motif Ontology Markup Language website. `motifml.org`.

[Schneiderman, 1998] Schneiderman, B. (1998). *Designing the User Interface*. Addison-Wesley, Reading, MA.

[Smith et al., 2002] Smith, M., McGuinness, D., Volz, R., and Welty, C. (2002). OWL website. `www.w3.org/TR/owl-guide/`.

[Tavazoie et al., 1999] Tavazoie, S., Hughes, J., Campbell, M., Cho, R., and Church, G. (1999). Systematic determination of genetic network architecture. *Nat. Genet.*, 22(3):281–285.

[W3C, 2001a] W3C (2001a). eXtensible Markup Language website. `www.w3.org/XML/`.

[W3C, 2001b] W3C (2001b). XML Schema website. `www.w3.org/XML/Schema`.

[W3C, 2001c] W3C (2001c). XML Stylesheet Language website. `www.w3.org/Style/XSL`.

[Wen et al., 1998] Wen, X., Fuhrman, S., Michaels, G., Carr, D., Smith, S., Barker, J., and Somogyi, R. (1998). Large-scale temporal gene expression mapping of central nervous system development. *Proc. Natl. Acad. Sci. U.S.A.*, 95(1):334–339.

[Yuh et al., 1998] Yuh, C., Bolouri, H., and Davidson, E. (1998). Genomic *cis*-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, 279:1896–1902.

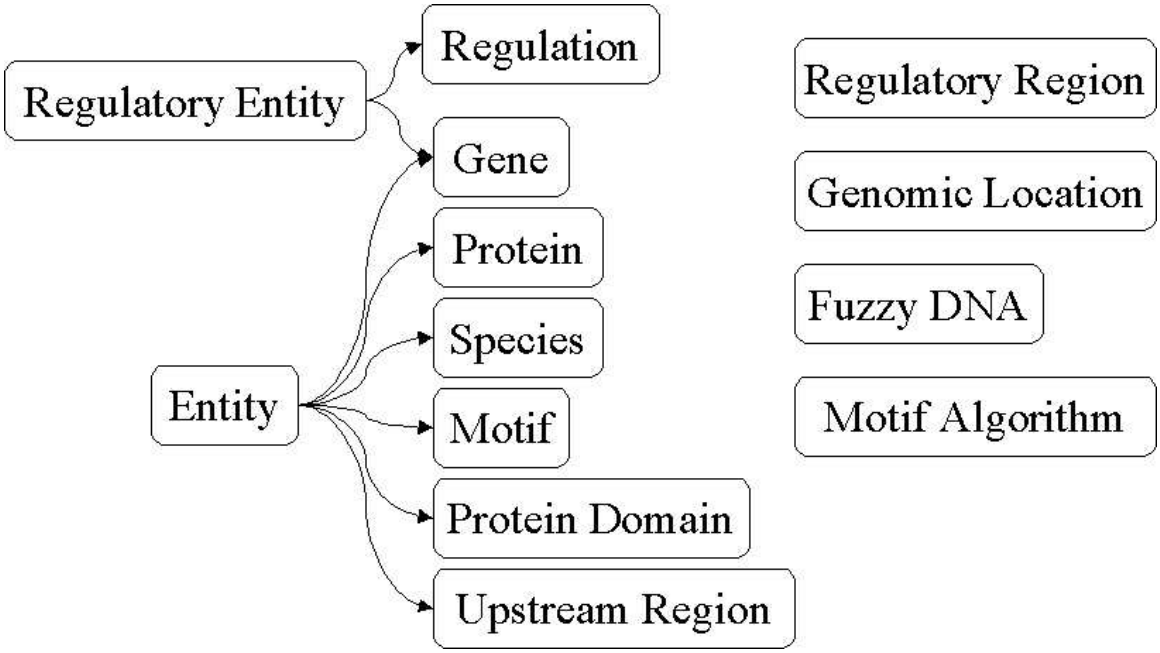

Figure 1: Top-level taxonomy of MotifML

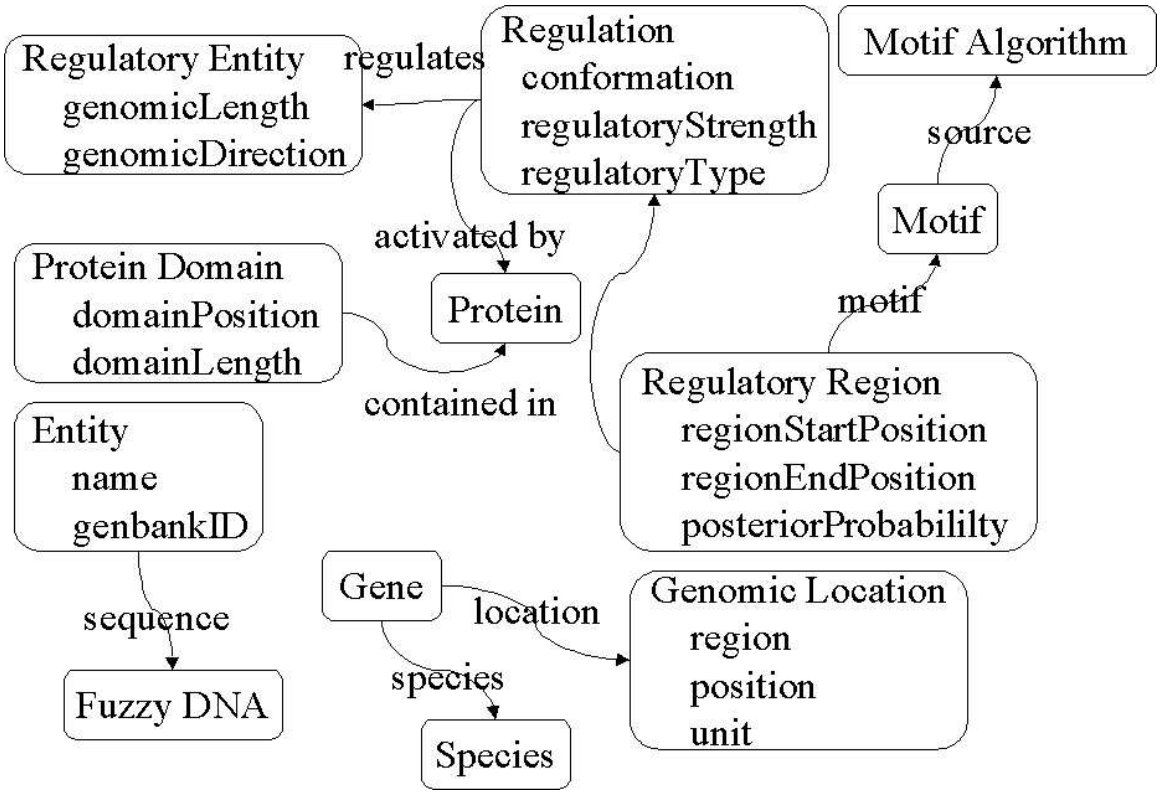

Figure 2: MotifML attributes and relationships

```
Motif #1:
*****************************
Width (14, 0); Gap [0, 0]; MotifScore 2.698; Segments 34;PValue 3.090950e-40

Blk1    A     G     C     T      Con  rCon Deg  rDeg
1     0.00  0.21  0.21  0.59     T    A    T    A
2     0.00  0.44  0.50  0.06     C    G    S    S
3     0.70  0.29  0.00  0.00     A    T    R    Y
4     0.32  0.62  0.00  0.06     G    C    R    Y
5     0.03  0.00  0.97  0.00     C    G    C    G
6     0.00  0.00  1.00  0.00     C    G    C    G
7     0.85  0.09  0.03  0.03     A    T    A    T
8     0.88  0.12  0.00  0.00     A    T    A    T
9     0.03  0.00  0.03  0.94     T    A    T    A
10    0.03  0.09  0.88  0.00     C    G    C    G
11    0.70  0.12  0.18  0.00     A    T    A    T
12    0.06  0.53  0.41  0.00     G    C    S    S
13    0.44  0.00  0.56  0.00     C    G    M    K
14    0.21  0.59  0.18  0.03     G    C    G    C

Seq #1  seg 1   r998    TCATCCAATCAGAG
Seq #2  seg 1   f91     TCAACCGAACAGAA
Seq #3  seg 1   r638    TCGACCAATCAAAA
Seq #4  seg 1   r168    TTAGCCAATCAGCA
Seq #5  seg 1   r259    TTAGCCAATCAGCA
Seq #6  seg 1   r588    CCAACCAATCCGAC
Seq #7  seg 1   r91     GCAGCCAGTCCCAG
Seq #8  seg 1   r238    GCAGCCAGTCCCAG
Seq #9  seg 1   r274    TGAGCCAATCACCG
Seq #9  seg 2   r358    TGGACCAATCAGAG
Seq #10 seg 1   f421    CGGACCGATCCGCC
Seq #11 seg 1   r206    CCAGCCAATCAGCC
Seq #12 seg 1   f107    TCAGCCAGCCGCCG
Seq #13 seg 1   r218    CCAGCCCATCAGAC
Seq #14 seg 1   r148    CCGGCCAATCGCCG
Seq #15 seg 1   r1491   CCAGACAGTCCCAA
Seq #16 seg 1   r274    TGAGCCAATCACCG
Seq #16 seg 2   r358    TGGACCAATCAGAG
Seq #17 seg 1   f1      GGATCCTATGAGCC
Seq #18 seg 1   r31     GCGGCCGATCAGCC
Seq #19 seg 1   r357    TGGACCAATCAGAG
Seq #19 seg 2   r273    TGAGCCAATCACCA
Seq #20 seg 1   r275    TGAGCCAATCACCG
Seq #20 seg 2   r359    TGGACCAATCAGAG
Seq #21 seg 1   r274    TGAGCCAATCACCG
Seq #21 seg 2   r358    TGGACCAATCAGAG
Seq #22 seg 1   r378    TCAGCCAATCACAA
Seq #23 seg 1   f1113   CCGGCCAATCGACG
Seq #24 seg 1   r1551   TCAGCCAATAGCAG
Seq #25 seg 1   r345    GGAACCAATCAGCG
Seq #25 seg 2   r262    TCAGCCAATGACCG
Seq #26 seg 1   r173    GGAGCCAATCCGCT
Seq #27 seg 1   r343    GGAACCAATCAGCG
Seq #27 seg 2   r261    TCAGCCAATGACCG
*****************************
```

Figure 3: Example of a motif in tabular format as produced by BioProspector

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
 xmlns:xsd ="http://www.w3.org/2001/XMLSchema#"
 xmlns:mml ="http://baclawski.com/motifml#">
<Motif rdf:ID="Motif1" sourceAlgorithm="#BioProspector">
  <sequence rdf:parseType="daml:collection">
    <FuzzyDNA C="0.21" G="0.21" T="0.59">
    <FuzzyDNA C="0.50" G="0.44" T="0.06">
    <FuzzyDNA A="0.70" G="0.29">
    <FuzzyDNA A="0.32" G="0.62" T="0.06">
    <FuzzyDNA A="0.03" C="0.97">
    <FuzzyDNA C="1.00">
    <FuzzyDNA A="0.85" C="0.03" G="0.09" T="0.03">
    <FuzzyDNA A="0.88" G="0.12">
    <FuzzyDNA A="0.03" C="0.03" T="0.94">
    <FuzzyDNA A="0.03" C="0.88" G="0.09">
    <FuzzyDNA A="0.70" C="0.18" G="0.12">
    <FuzzyDNA A="0.06" C="0.41" G="0.53">
    <FuzzyDNA A="0.44" C="0.56">
    <FuzzyDNA A="0.21" C="0.18" G="0.59" T="0.03">
  </sequence>
</Motif>
<RegulatoryRegion regionStartPosition="-998" genomicDirection="#reverse">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene1"/>
  <sequence>TCATCCAATCAGAG</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-91" genomicDirection="#forward">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene2"/>
  <sequence>TCAACCGAACAGAA</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-638" genomicDirection="#reverse">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene3"/>
  <sequence>TCGACCAATCAAAA</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-168" genomicDirection="#reverse">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene4"/>
  <sequence>TTAGCCAATCAGCA</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-259" genomicDirection="#reverse">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene5"/>
  <sequence>TTAGCCAATCAGCA</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-588" genomicDirection="#reverse">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene6"/>
  <sequence>CCAACCAATCCGAC</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-91" genomicDirection="#reverse">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene7"/>
  <sequence>GCAGCCAGTCCCAG</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-238" genomicDirection="#reverse">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene8"/>
  <sequence>GCAGCCAGTCCCAG</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-274" genomicDirection="#reverse">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene9"/>
  <sequence>TGAGCCAATCACCG</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-358" genomicDirection="#reverse">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene9"/>
  <sequence>TGGACCAATCAGAG</sequence>
</RegulatoryRegion>
<RegulatoryRegion regionStartPosition="-421" genomicDirection="#forward">
  <motif rdf:resource="Motif1"/>
  <regulates rdf:resource="Gene10"/>
  <sequence>CGGACCGATCCGCC</sequence>
</RegulatoryRegion>
```

Figure 4: MotifML representation of a motif

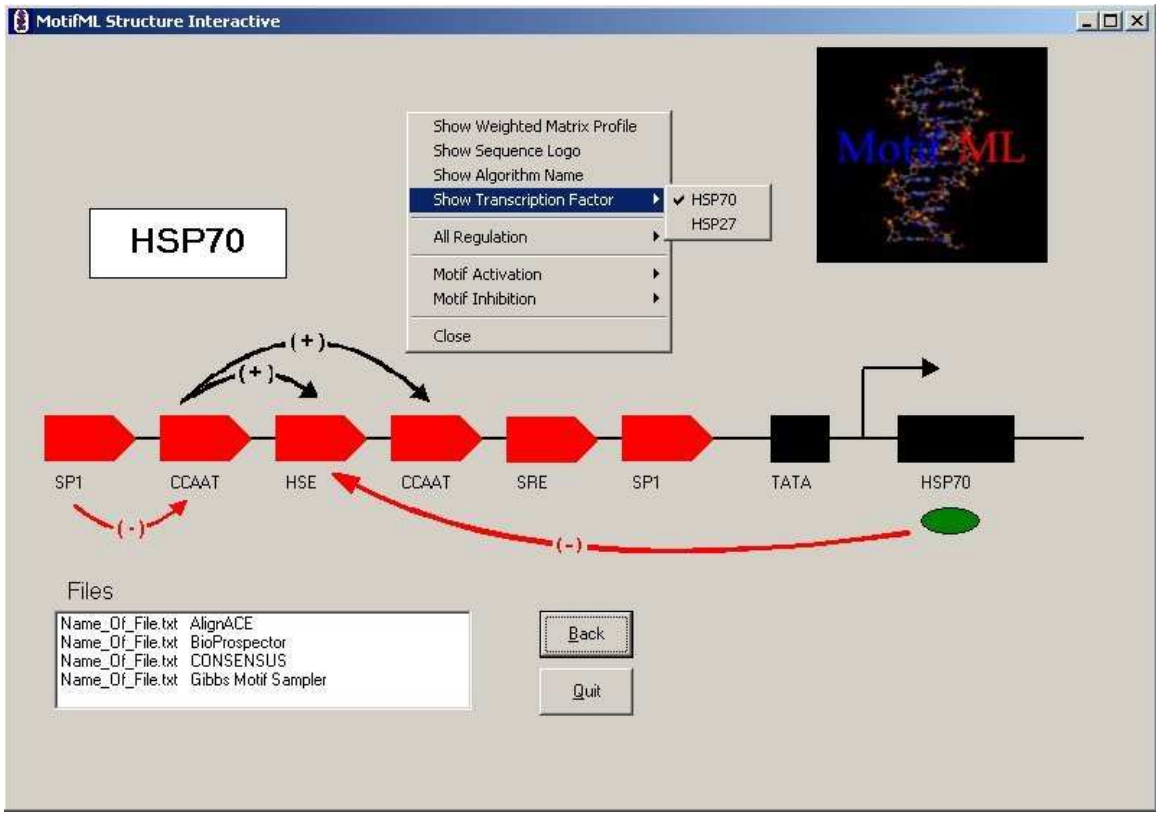

Figure 5: Example of the MotifML Graphic User Interface

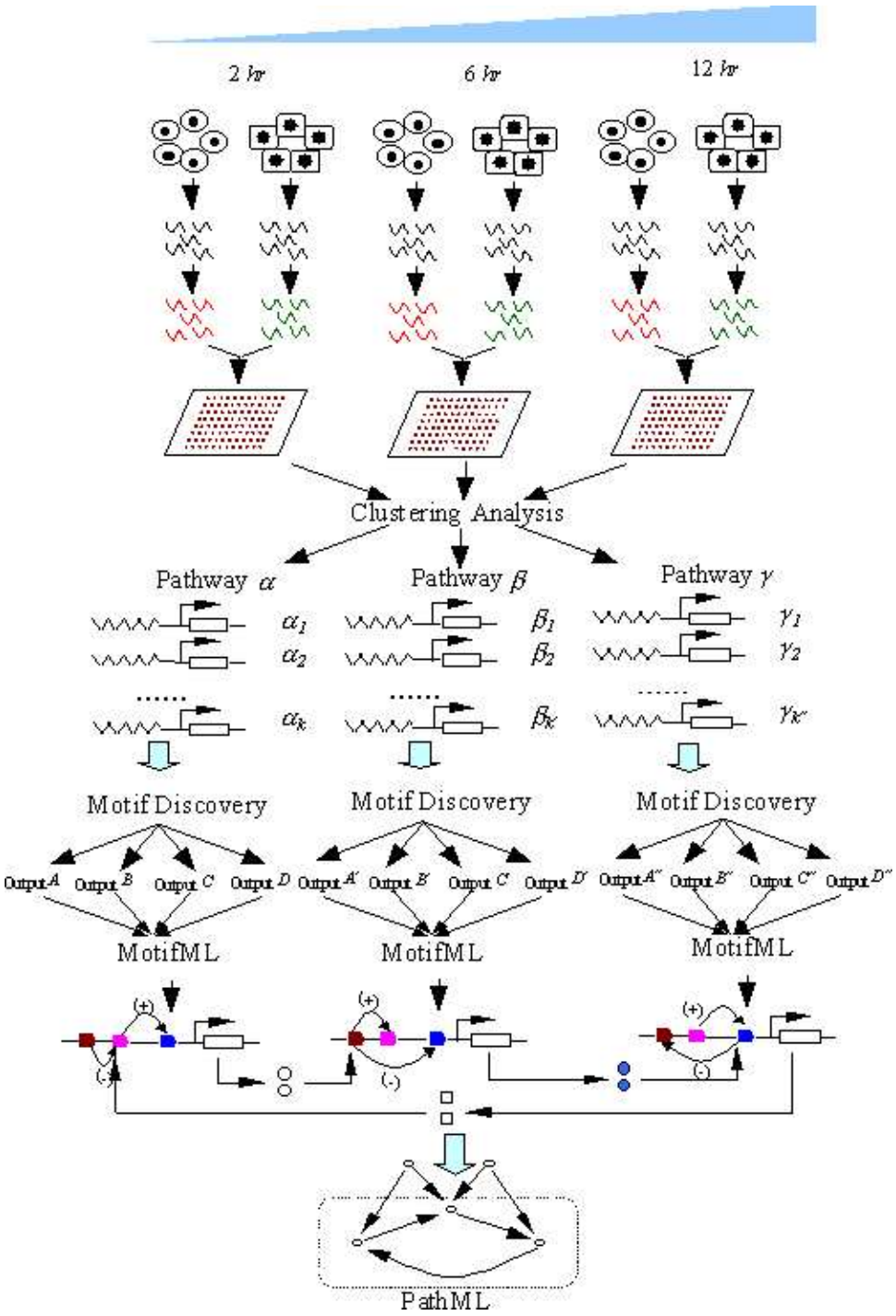

Figure 6: Knowledge Mining of the Genetic Regulatory Circuitry