Chapter 5
Mining Spatial Association Rules for Composite Motif Discovery

Michelangelo Ceci, Corrado Loglisci, Eliana Salvemini, Domenica D'Elia, and Donato Malerba

Abstract Motif discovery in biological sequences is an important field in bioinformatics. Most of the scientific research focuses on the de novo discovery of single motifs, but biological activities are typically co-regulated by several factors and this feature is properly reflected by higher order structures, called composite motifs, or cis-regulatory modules or simply modules. A module is a set of motifs, constrained both in number and location, which is statistically overrepresented and hence may be indicative of a biological function. Several methods have been studied for the de novo discovery of modules. We propose an alternative approach based on the discovery of rules that define strong spatial associations between single motifs and suggest the structure of a module. Single motifs involved in the mined rules might be either de novo discovered by motif discovery algorithms or taken from databases of single motifs. Rules are expressed in a first-order logic formalism and are mined by means of an inductive logic programming system. We also propose computational solutions to two issues: the hard discretization of numerical inter-motif distances and the choice of a minimum support threshold. All methods have been implemented and integrated in a tool designed to support biologists in the discovery and characterization of composite motifs. A case study is reported in order to show the potential of the tool.

5.1 Introduction

In biological sequence analysis, a *motif* is a nucleotide or amino-acid sequence pattern which appears in a set of sequences (DNA, RNA or protein) with much higher frequency than would be expected by chance. This statistical overrepresentation is

D. Malerba (✉)
Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro," via Orabona 4, 70126 Bari, Italy
e-mail: malerba@di.uniba.it

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_5,

expected to be indicative of an associated biological function. Examples of motifs include DNA- and RNA-binding sites for regulatory proteins, protein domains and protein epitopes.

DNA and RNA motifs are key to deciphering the language of gene regulatory mechanisms and, in particular, to fully understand how gene expression is regulated in time and space. For this reason, de novo (or ab initio) motif discovery, i.e. identifying motif sites (signals) in a given set of unaligned biological sequences, has attracted the attention of many biologists. However, they are also difficult to identify, since motifs often produce weak signals buried in genomic noise (i.e. the background sequence) [8]. This problem is known to be NP-hard [22], thus it is also an interesting arena for computer scientists.

Most of the motif discovery tools reported in the literature are designed to discover single motifs. However, in many (if not most) cases, biological activities are co-regulated by several factors. For instance, transcription factor-binding sites (TFBSs) on DNA are often organized in functional groups called *composite motifs* or *cis-regulatory modules* (CRM) or simply modules. These modules may have a biologically important structure that constrains both the number and relative position of the constituent motifs [34].

One example, among many that could be cited, is ETS-CBF, a cis-regulatory module constituted by three single motifs, μA, μB and CBF (core-binding factor). Both μA and μB are binding sites for two transcription factors belonging to the ETS proteins family, Ets-1 and PU.1, respectively. CBF is a protein that is implicated in the activation of several T and myeloid cell-specific promoters and enhancers. Enhancers are cis-regulatory sequences which control the efficiency of gene transcription from an adjacent promoter. ETS-CBF is a common composite motif of enhancers implicated in the regulation of antigen receptor genes in mouse and human. A comparative study of the tripartite domain of the murine immunoglobulin μ heavy-chain (IgH) enhancer and its homologous in human has demonstrated that in both species the activity of the gene enhancer is strictly dependent on ETS-CBF [12].

Therefore, it is of great interest to discover not only single motifs but also the higher order structure into which motifs are organized, i.e. the modules. This problem is also known as *composite* [38] or *structured* [32] *motif* discovery.

Over the past few years, a plethora of single motif discovery tools have been reported in the literature (see the book by Robin et al. [36]). They differ in three aspects:

1. The representation of a pattern that constitutes a single motif,
2. The definition of overrepresentation of a motif pattern and
3. The search strategy applied to pattern finding.

A single motif can be represented either by a consensus sequence, which contains the most frequent nucleotide in each position of the observed signals, or by a position weight matrix (PWM), which assigns a different probability to each possible letter at each position in the motif [46].

Both consensus sequences and PWMs are derived by the multiple alignments of all the known recognition sites for a given regulatory factor and represent the specificity of a regulatory factor for its own recognition site. They refer to a sequence that matches all the sequences of the aligned recognition sites very closely, but not necessarily exactly. In a consensus sequence, this concept is expressed by notations that indicate which positions of the consensus sequence are always occupied by the same nucleotide (exact match) and which one can vary and how (allowed mismatch), without affecting the functionality of the motif. Considering the example DNA consensus sequence T[CT]NG{A}A, it has to be read in the following way: the first, fourth and sixth position in the consensus are always occupied by T, G and A, where T stands for thymidine, G for guanine and A for adenine; no mismatches are allowed in these positions. The second nucleotide in the sequence can be a cytosine (C) or alternatively a T. This mismatch does not affect the effectiveness of the recognition signal. The third position of the consensus can be occupied by any of the four nucleotide bases (A, T, C, G). At the fifth position any base can be present except A.

A PWM of a DNA motif has one row for each nucleotide base (A, T, C, G) and one column for each position in the pattern. This way, there is a matrix element for all possible basis at every position. The score of any particular motif in a sequence of DNA is the sum of the matrix values for that motif's sequence. This score is the same of the consensus only when the motif perfectly matches the consensus. Any sequence motif that differs from the consensus in some positions will have a lower score depending on the number and type of nucleotide mismatches.

In contrast to these sequence patterns, spatial patterns have also been investigated [19], where spatial relationships (e.g. adjacency and parallelism) and shapes (e.g., α-helices in protein motifs) can be represented.

The overrepresentation of motif patterns has been defined in several ways. In some motif-discovery algorithms, a score is defined for each pattern (e.g., p-value [47] or z-score [43]), and the observed motif scores are compared with expected scores from a background model. In other algorithms, two separate values are computed when evaluating motifs, one concerning the support, or coverage, of a motif, and the other concerning the unexpectedness of a motif [35]. A third approach is to use a measure of information content [25] of discovered patterns.

Search strategies can be categorized as enumerative (or pattern-driven) and heuristic (or sequence-driven). The former enumerate all possible motifs in a given solution space (defined by a template pattern) and test each for significance, while the latter try to build a motif model by varying some model parameters such that a matching score with sequence data is maximized. In general, enumerative algorithms find optimal solutions for discrete representations of relatively short motifs, but do not scale well to larger motifs and continuous models. TEIRESIAS [35] is more sophisticated in using information about the relative occurrences of substrings; therefore, it can be used to discover discrete representations of longer motifs. Among the heuristic-based approaches, the most common is the expectation-maximization (EM) [5], which is a deterministic local search algorithm. EM may converge very fast, but the optimality of the returned point strongly depends on

the starting point (seed). For this reason, it is used in combination with some randomization techniques in order to escape from a poor local optimum even if the chosen seed is bad [6].

Algorithms for the de novo discovery of modules, together with the parameters of their constituent motifs [14, 41, 52], are more recent. These algorithms, which exploit some form of spatial information (e.g., spatial correlation) on constituent motifs to identify a module, are considered particularly promising since they may offer both improved performance over conventional discovery algorithms of single motifs and insight into the mechanism of regulation directed by the constituent motifs [26]. However, in order to restrict the search space, they make some assumptions which limit their flexibility in handling variations of either the number or length of the constituent motifs or the spacing between them. For instance, the hierarchical mixture (HMx) model of CISMODULE [52] requires the specification of both the length of the module and the total number of constituent motif types. Moreover, CISMODULE does not capture the order or precise spacing of multiple TFBSs in a module. Segal and Sharan [41] propose a method for the de novo discovery of modules consisting of a combination of single motifs which are spatially close to each other. Despite the flexibility of their method in handling modules, they assume that a training set (with positive and negative examples of transcriptional regulation) is available in order to learn a discriminative model of modules. The method EMC module proposed by Gupta and Liu [14] assumes a geometrical probability distribution on the distance between TFBSs.

Although a recent study [18] has shown a significant improvement in prediction success when modules are considered instead of isolated motifs, it is largely believed that without some strong form of inductive bias,[1] methods for de novo module discovery may have performance close to random. For this reason, another line of module discovery methods has been investigated (e.g., Cister [13], ModuleSearcher [1], MScan [20], Compo [39]), which takes a list of single motifs as input along with the sequence data in which the modules should be found. Single motifs are taken from motif databases, such as TRANSFAC [15] and JASPAR [37], and the challenges concern discovering which of them are involved in the module, defining the sequence of single motifs in the module and possibly discovering the inter-motif distances.[2]

Module discovery methods can be categorized according to the type of framework, either discrete (e.g., CREME [42]) or probabilistic (e.g., Logos [51]), adopted to model modules. In a discrete framework, all constituent motifs must appear in a module instance. This simplifies inference and interpretation of modules, and often allows exhaustive search of optimal constituent motifs in a sequence window

[1] The inductive bias of a learning algorithm is the set of assumptions that the learner uses to predict outputs given inputs that it has not encountered. It forms the rationale for learning since without it no generalization is possible [29].

[2] The distance is typically evaluated as the number of nucleotides which separate two consecutive single motifs. More sophisticated distance measures might be used in future works if significant progress is made in the prediction of DNA folding.

of a given length. Conversely, a probabilistic framework is more expressive, since it relaxes the hard constraints of discrete frameworks and associates each module with a score which is a combination (e.g., the sum) of motifs and distance scores. Issues of probabilistic frameworks are local optima and interpretability of results.

A recent assessment of eight published methods for module discovery [21] has shown that no single method performed consistently better than others in all situations and that there are still advances to be made in computational module discovery. In this chapter, we propose an innovative approach to module discovery, which can be a useful supplement or alternative to other well-known approaches. The idea is to mine rules which define "strong" spatial associations between single motifs [27]. Single motifs might either be de novo discovered by traditional discovery algorithms or taken from databases of known motifs.

The spatial relationships considered in this work are the order of motifs along the DNA sequence and the inter-motif distance between each consecutive couple of motifs, although the mining method proposed to generate spatial association rules has no limitation on both the number and the nature of spatial relationships. The association rule mining method is based on an inductive logic programming (ILP) [31] formulation according to which both data and discovered patterns are represented in a first-order logic formalism. This formulation also facilitates the accommodation of diverse sources of domain (or background) knowledge which are expressed in a declarative way. Indeed, ILP is particularly well suited to bioinformatics tasks due to its ability both to take into account background knowledge and to work directly with structured data [30]. This is confirmed by some notable success in molecular biology applications, such as predicting carcinogenesis [44, 45].

The proposed approach is based on a discrete framework, which presents several advantages, the most relevant being the straightforward interpretation of rules, but also some disadvantages, such as the hard discretization of numerical inter-motif distances or the choice of a minimum support threshold. To overcome these issues, some computational solutions have been developed and tested.

The specific features of this approach are:

- An original perspective of module discovery as a spatial association rule mining task;
- A logic-based approach where background knowledge can be expressed in a declarative way;
- A procedure for the automated selection of some parameters which are difficult to properly set;
- Some computational solutions to overcome the discretization issues of discrete approaches.

These features provide our module discovery tool several advantages with respect to competitive approaches. First, spatial association rules, which take the form of $A \Rightarrow C$, provide insight both into the support of the module (represented by $A \wedge C$) and into the confidence of possible predictions of C given A. Predictions may equally concern both properties of motifs (e.g., its type) and spatial relationships (e.g., the inter-motif distance). Second, the declarative knowledge

representation facilitates the development and debugging of background knowledge in collaboration with a domain expert. Moreover, knowledge expressed in a declarative way is re-usable across different tasks and domains, thus easing the burden of the knowledge engineering effort. Third, the resort to first-order (or relational) logic facilitates the representation of input sequences, whose structure can be arbitrarily complex, and increases the explanatory power of discovered patterns, which are relatively easy to interpret for domain experts. Fourth, computational solutions devised for both the problem of selecting a minimum support threshold and the problem of discretizing numerical data fulfill the twofold goal of improving the quality of results and designing tools for the actual end-users, namely biologists.

Further significant advantages are:

- No prior assumption is necessary either on the constituent motifs of a module or on their spatial distribution;
- Specific information on the bases occurring between two consecutive motifs is not required.

This work also extends our previous study [48], where frequent patterns are generated by means of the algorithm GSP [3]. The extension aims to: (1) find association rules, which convey additional information with respect to frequent patterns; (2) discover more significant inter-motif distances by means of a new discretization algorithm which does not require input parameters; (3) automatically select the best minimum support threshold; (4) filter redundant rules; (5) investigate a new application of an ILP algorithm to a challenging bioinformatics task.

The chapter is organized as follows. Section 5.2 presents a formalization of the problem, which is decomposed into two subproblems: (1) mining frequent sets of motifs, and (2) mining spatial association rules. Input and output of each step of the proposed approach are also reported. Section 5.3 describes the method for spatial association rule mining. Section 5.4 presents the solution to some methodological and architectural problems which affect the implementation of a module discovery tool effectively usable by biologists. Section 5.5 is devoted to a case study, which shows the application of the developed system. Finally, conclusions are drawn.

5.2 Mining Spatial Association Rules from Sequences

Before proceeding to a formalization of the problem, we first introduce some general notions on association rules.

Association rules are a class of patterns that describe regularities or co-occurrence relationships in a set T of homogeneous data structures (e.g., sets, sequences and so on) [2]. Formally, an association rule R is expressed in the form of $A \Rightarrow C$, where A (the *antecedent*) and C (the *consequent*) are disjoint conditions on properties of data structures (e.g., the presence of an item in a set). The meaning of an association rule is quite intuitive: if a data structure satisfies A, then it is likely to satisfy C. To quantify this likelihood, two statistical parameters are usually

computed, namely *support* and *confidence*. The former, denoted as $sup(R, T)$, estimates the probability $P(A \wedge C)$ by means of the percentage of data structures in T satisfying both A and C. The latter, denoted as $conf(R, T)$, estimates the probability $P(C|A)$ by means of the percentage of data structures which satisfy condition C, out of those which satisfy condition A. The task of association rule mining consists in discovering all rules whose support and confidence values exceed two respective minimum thresholds. When data structures describe spatial objects together with their spatial relationships, mined association rules are called *spatial*, since conditions in either the antecedent or the consequent of a rule express some form of spatial constraint.

We now give a formal statement of the module discovery problem, which is decomposed into two subproblems as follows:

1. *Given:* A set M of single motifs, a set T of sequences with annotations about type and position of motifs in M and a minimum value τ_{min},
   Find: The collection $\mathcal{S}$ of all the sets $S_1, S_2, \ldots, S_n$ of single motifs such that, for each S_i, at least τ_{min} sequences in T contain all motifs in S_i.
2. *Given:* A set $S \in \mathcal{S}$ and two thresholds σ_{min} and κ_{min},
   Find: Spatial association rules involving motifs in S, such that their support and confidence are greater than σ_{min} and κ_{min}, respectively.

Single motifs in M can be either discovered de novo or taken from a single motif database. Each $S_i \in \mathcal{S}$ is called *motif set*. The *support set* of S_i is the subset T_{S_i} of sequences in T such that each sequence in T_{S_i} contains at least one occurrence of each motif in S_i. According to the statement of subproblem (1) $|T_{S_i}| \geq \tau_{min}$. T_{S_i} is used to evaluate both support and confidence of spatial association rules mentioned in subproblem (2).

The proposed approach is two-stepped since it reflects this problem decomposition. In the first step, motif sets which are *frequent*, i.e., have a support greater than τ_{min}, are extracted from sequences annotated with predictions for known single motifs. Only information about the occurrence of motifs is considered, while spatial distribution of motifs is ignored. This step has a manifold purpose: (1) enabling biologists to guide deeper analysis only for sets of motifs which are deemed potentially interesting; (2) filtering out sequences which do not include those interesting sets of motifs; (3) lowering the computational cost of the second step.

In the second step, sequences that support specific frequent motif sets are abstracted into *sequences of spaced motifs*. A sequence of spaced motifs is defined as an ordered collection of motifs interleaved with inter-motif distances. Each inter-motif distance measures the distance between the last nucleotide of a motif and the first nucleotide of the next motif in the sequence. Spatial association rules are mined from these abstractions. In order to deal with numerical information on the inter-motif distance, a discretization algorithm is applied. The algorithm takes into account the distribution of the examples and does not significantly depend on input parameters as in the case of classical equal width or equal frequency discretization algorithms. Details on both steps are reported below.

5.2.1 Mining Frequent Motif Sets

To solve the first sub-problem, we resort to the levelwise breadth-first search [28] in the lattice of motif sets. The search starts from the smallest element, i.e., sets with one motif in M, and proceeds from smaller to larger sets. The frequent sets of i motifs are used to generate candidate sets of $(i + 1)$ motifs. Each candidate set, which is potentially frequent, is evaluated against the set T of sequences, in order to check the actual size of its support set. Then it is pruned if its support is lower than τ_{min}. For instance, given $M = \{x, y, z\}$ and T as in Fig. 5.1a, the set $S = \{x, y\}$ is supported by $T_S = \{t_2, t_3\}$. If $\tau_{min} = 2$, then S is returned together with other frequent motif sets in $\mathcal{S}$.

5.2.2 Mining Spatial Association Rules

The sequences in the support set T_S of a frequent motif set S are represented as chains of the form $\langle m_1, d_1, m_2, \ldots, d_{n-1}, m_n \rangle$, where each m_i denotes a single motif ($m_i \in M$), while each d_i, $i = 1, 2, \ldots, n-1$, denotes the inter-motif distance between m_i and m_{i+1}. Each chain is a sequence of spaced motifs. For instance, sequence t_2 in Fig. 5.1a is represented as $\langle x, 10, y, 92, y \rangle$.

From a biological viewpoint, slight differences in inter-motif distances can be ignored. For this reason, we can group almost equal distances by applying a discretization technique which maps numerical values into a set of closed intervals.

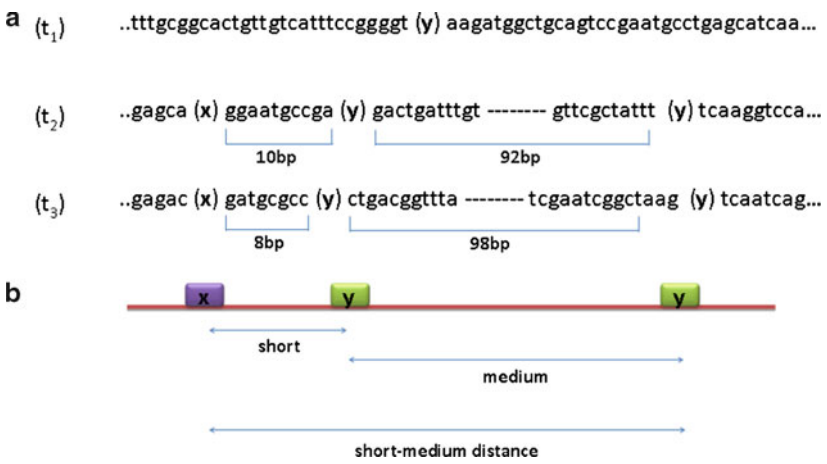

Fig. 5.1 (**a**) Three different annotated sequences (t_1, t_2, t_3) belonging to the set T where motifs x and y have been found. The *grey* semi-boxes underline the nucleotide sequences between two consecutive motifs (inter-motif distance). Inter-motif distances are expressed in base pairs (bp). (**b**) Closed-intervals of inter-motif distances

Therefore, a sequence of spaced motifs can be further abstracted into an ordered collection of motifs interleaved by symbols (e.g., short, medium, and large) representing a range of inter-motif distance. For instance, by considering the closed intervals in Fig. 5.1b, both sequences t_2 and t_3 in Fig. 5.1a are represented by the following sequence of spaced motifs:

$$\langle x, short, y, medium, y \rangle. \tag{5.1}$$

Each sequence of spaced motifs is described in a logic formalism which can be processed by the ILP system SPADA (**S**patial **Pa**ttern **D**iscovery **A**lgorithm) [24] to generate spatial association rules. More precisely, the whole sequence, the constituent motifs and the inter-motif distances are represented by distinct constant symbols.[3] Some predicate symbols are introduced in order to express both properties and relationships. They are:

- *sequence(t)*: t is a sequence of spaced motifs;
- *part_of(t,m)*: The sequence t contains an occurrence m of single motif;
- *is_a(m,x)*: The occurrence m is a motif x;
- *distance(m_1,m_2,d)*: The distance between the occurrences m_1 and m_2 is d.

A sequence is represented by a set of *Datalog*[4] *ground atoms*, where a Datalog ground atom is an n-ary predicate symbol applied to n constants. For instance, the sequence of spaced motif in (5.1) is described by the following set of Datalog ground atoms:

$$\left\{ \begin{array}{c} sequence(t_2), \\ part\_of(t_2, m_1),\ part\_of(t_2, m_2),\ part\_of(t_2, m_3), \\ is\_a(m_1, x),\ is\_a(m_2, y),\ is\_a(m_3, y), \\ distance(m_1, m_2, short),\ distance(m_2, m_3, medium). \end{array} \right\} \tag{5.2}$$

The set of Datalog ground atoms of all sequences is stored in the *extensional* part D_E of a deductive database D. The *intensional* part D_I of the deductive database D includes the definition of the domain knowledge in the form of *Datalog rules*. An example of Datalog rules is the following:

$$\begin{array}{c} short\_medium\_distance(U, V) \leftarrow distance(U, V, short). \\ short\_medium\_distance(U, V) \leftarrow distance(U, V, medium). \end{array} \tag{5.3}$$

They state that two motifs[5] are at a *short_medium_distance* if they are at either *short* or *medium* distance (Fig. 5.1b). Rules in D_I allows additional Datalog

[3] We denote constants as strings of lowercase letters possibly followed by subscripts.

[4] Datalog is a query language for deductive databases [9].

[5] Variables are denoted by uppercase letters possibly followed by subscripts, such as U and V.

ground atoms to be deduced from data stored in D_E. For instance, rules in (5.3) entail the following information from the set of Datalog ground atoms in (5.2):

$$\left\{ \begin{array}{l} short\_medium\_distance(m_1, m_2), \\ short\_medium\_distance(m_2, m_3). \end{array} \right\} \tag{5.4}$$

SPADA adds these entailed Datalog ground atoms to set (5.2), so that atoms with the predicate *short_medium_distance* can also appear in mined association rules.

Spatial association rules discovered by SPADA take the form $A \Rightarrow C$, where both A and C are conjunctions of *Datalog non-ground atoms*. A Datalog ground atom is an n-ary predicate symbol applied to n terms (either constants or variables), at least one of which is a variable. For each association rule, there is exactly one variable denoting the whole sequence and other variables denoting constituent motifs. An example of a spatial association rule is the following:

$$\begin{array}{l} sequence(T), part\_of(T, M_1), is\_a(M_1, x), distance(M_1, M_2, short), \\ \quad M_1 \neq M_2 \Rightarrow is\_a(M_2, y) \end{array} \tag{5.5}$$

where variable T denotes a sequence, while variables M_1 and M_2 denote two distinct occurrences of single motifs ($M_1 \neq M_2$) of type x and y, respectively. With reference to the sequence described in (5.2), T corresponds to t_2 while the two distinct occurrences of single motifs M_1 and M_2 correspond to m_1 and m_2, respectively. By means of this association rule, it is possible to infer which is the single motif that follows in a short distance a single motif x. The uncertainty of the inference is quantified by the confidence of the association rule.

Details on the association rule discovery algorithm implemented in SPADA are reported in the next section.

5.3 SPADA: Pattern Space and Search Procedure

In SPADA, the set O of spatial objects is partitioned into a set S of *reference* (or target) *objects* and m sets R_k, $1 \leq k \leq m$, of *task-relevant* (or non-target) objects. Reference objects are the main subject of analysis and contribute to the computation of the support of a pattern, while task-relevant objects are related to the reference objects and contribute to accounting for the variation, i.e., they can be involved in a pattern. In the sequence described in (5.2), the constant t_2 denotes a reference object, while the constants m_1, m_2 and m_3 denote three task relevant objects. In this case, there is only one set R_1 of task-relevant objects.

SPADA is the only ILP system which addresses the task of relational frequent pattern discovery by dealing properly with concept hierarchies. Indeed, for each set R_k, a generalization hierarchy H_k is defined together with a function ψ_k, which maps objects in H_k into a set of granularity levels $\{1, \ldots, L\}$. For instance, with

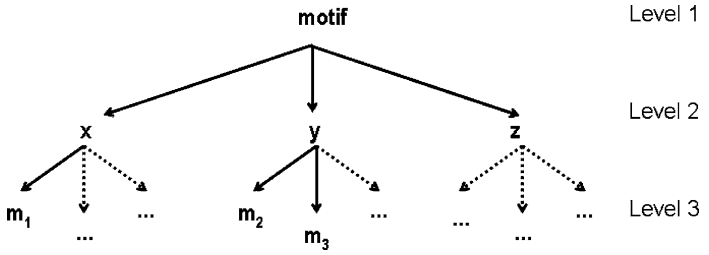

Fig. 5.2 A three-level hierarchy defined on motifs

reference to the sequence described in (5.2), it is possible to define a three-level hierarchy H_1 (Fig. 5.2), where the top level represents a generic single motif, the middle level represents distinct single motifs in M and the lowest level represents specific occurrences of motifs. In this example, the function ψ_1 simply maps the root to 1, x, y, and z to 2 and m_1, m_2 and m_3 to 3.

The set of predicates used in SPADA can be categorized into four classes. The *key predicate* identifies the reference objects in S (e.g., *sequence* is the key predicate in description (5.2)). The *property predicates* are binary predicates which define the value taken by an attribute of an object (e.g., *length* of a motif, not reported in description (5.2)). The *structural predicates* are binary predicates which relate task-relevant objects (e.g., *distance*) as well as reference objects with task-relevant objects (e.g., *part_of*). The *is_a* predicate is a binary *taxonomic* predicate which associates a task-relevant object with a value of some H_k.

The *units of analysis* $D[s]$, one for each reference object $s \in S$, are subsets of ground facts in D_E, defined as follows:

$$D[s] = is\_a(R(s)) \cup D[s|R(s)] \cup \bigcup_{r_i \in R(s)} D[r_i|R(s)], \tag{5.6}$$

where:

- $R(s)$ is the set of task-relevant objects directly or indirectly related to s;
- $is\_a(R(s))$ is the set of *is_a* atoms specified for each $r_i \in R(s)$;
- $D[s|R(s)]$ contains both properties of s and relations between s and some $r_i \in R(s)$;
- $D[r_i|R(s)]$ contains both properties of r_i and relations between r_i and some $r_j \in R(s)$.

This notion of unit of analysis is coherent with the individual-centered representation [7], which has some nice properties, both theoretical (e.g., PAC-learnability [49]) and computational (e.g., smaller hypothesis space and more efficient search). The set of units of analysis is a partitioning of D_E into a number of subsets $D[s]$, each of which includes ground atoms concerning the task-relevant objects (transitively) related to the reference object s. With reference to the sequence described in (5.2), $R(t_2) = \{m_1, m_2, m_3\}$, and $D[t_2]$ coincides with the whole set of ground

atoms, including those inferred by means of rules in the intensional part D_I of the deductive database. If several reference objects had been reported in (5.2), $D[t_2]$ would have been a proper subset.

Patterns discovered by SPADA are conjunctions of Datalog non-ground atoms, which can be expressed by means of a set notation. For this reason they are also called *atomsets* [10], by analogy with itemsets introduced for classical association rules. A formal definition of atomset is reported in the following.

Definition 5.1. An *atomset* P is a set of atoms $p_0(t_0^1)$, $p_1(t_1^1, t_1^2)$, $p_2(t_2^1, t_2^2)$, ..., $p_r(t_r^1, t_r^2)$, where p_0 is the key predicate, while p_i, $i = 1, \ldots, r$, is either a structural predicate or a property predicate or an *is_a* predicate.

Terms t_i^j are either constants, which correspond to values of property predicates, or variables, which identify reference objects either in S or in some R_k. Each p_i is a predicate occurring either in D_E (extensionally defined predicate) or in D_I (intensionally defined predicate). Some examples of atomsets are the following:

$$P_1 \equiv sequence(T), part\_of(T, M_1), is\_a(M_1, x)$$

$$P_2 \equiv sequence(T), part\_of(T, M_1), is\_a(M_1, x), distance(M_1, M_2, short)$$

$$P_3 \equiv sequence(T), part\_of(T, M_1), is\_a(M_1, x), distance(M_1, M_2, short), is\_a(M_2, y)$$

where variable T denotes a reference object, while variables M_1 and M_2 denote some task-relevant objects. All variables are implicitly existentially quantified.

Atomsets in the search space explored by SPADA satisfy the *linkedness* [16] property, which means that each variable denoting a task-relevant object in an atomset P defined as in Definition 5.1 must be transitively linked to the reference object t_0^1 by means of structural predicates. For instance, variables M_1 and M_2 in P_1, P_2 and P_3 are transitively linked to T by means of the structural predicates *distance* and *part_of*. Therefore, P_1, P_2 and P_3 satisfy the linkedness property.

Each atomset P is associated with a granularity level l. This means that all taxonomic (*is_a*) atoms in P refer to task-relevant objects, which are mapped by some ψ_k into the same granularity level l. For instance, atomsets P_1, P_2 and P_3 are associated with the granularity level 2 according to the hierarchy H_1 in Fig. 5.2 and the associated function ψ_1. For the same reason, the following atomset:

$$P_4 \equiv sequence(T), part\_of(T, M_1), is\_a(M_1, motif)$$

is associated with the granularity level 1.

In multi-level association rule mining, it is possible to define an *ancestor* relation between two atomsets P and P' at different granularity levels.

Definition 5.2. An atomset P at granularity level l is an *ancestor* of an atomset P' at granularity level l', $l < l'$, if P'can be obtained from P by replacing each

task-relevant object $h \in H_k$ at granularity level l ($l = \psi_k(h)$) with a task-relevant object h', which is more specific than h in H_k and is mapped into the granularity level l' ($l' = \psi_k(h')$).

For instance, the atomset P_4 defined above is an ancestor of P_1, since P_1 can be obtained from P_4 by replacing *motif* with x.

By associating an atomset P with an existentially quantified conjunctive formula *eqc(P)* obtained by transforming P into a Datalog query, we can now provide a formal definition of the support of P on a deductive database D. We recall that D has an extensional part D_E and an intensional part D_I. Moreover D_E includes several units of analysis $D[s]$ one for each reference object.

Definition 5.3. An atomset P *covers* a unit of analysis $D[s]$ if $D[s] \cup D_I$ logically entails $eqc(P)$ ($D[s] \cup D_I \models eqc(P)$).

Each atomset P is associated with a support, denoted as *sup(P,D)*, which is the percentage of units of analysis in D covered by P. The minimum support for frequent atomsets depends on the granularity level l of task-relevant objects. It is denoted as $\sigma_{min}[l]$ and we assume that $\sigma_{min}[l+1] \leq \sigma_{min}[l]$, $l = 1, 2, \ldots, L$-1.

Definition 5.4. An atomset P at granularity level l with support *sup(P,D)* is *frequent* if $sup(P, D) \geq \sigma_{min}[l]$ and all ancestors of P are frequent at their corresponding levels.

In SPADA, the discovery of frequent atomsets is performed according to both an intra-level and an inter-level search. The intra-level search explores the space of patterns at the same level of granularity. It is based on the level-wise method [28], which performs a breadth-first search of the space, from the most general to the most specific patterns, and prunes portions of the search space which contain only infrequent patterns.

The application of the level-wise method requires a generality ordering, which is monotonic with respect to pattern support. The generality ordering adopted by SPADA is based on the notion of θ-subsumption [33].

Definition 5.5. P_1 is more general than P_2 under θ-subsumption ($P_1 \succeq_\theta P_2$) if and only if P_1 θ-subsumes P_2, i.e., a substitution θ exists, such that $P_1\theta \subseteq P_2$.

For instance, with reference TO the atomsets P_1, P_2 and P_3 reported above, we observe that P_1 θ-subsumes P_2 ($P_1 \succeq_\theta P_2$) and P_2 θ-subsumes P_3 ($P_2 \succeq_\theta P_3$) with substitutions $\theta_1 = \theta_2 = \oslash$.

The relation $\succeq_\theta$ is a quasi-ordering (or preorder), since it is reflexive and transitive but not antisymmetric. Moreover, it is monotonic with respect to support [24], as stated in the following proposition.

Proposition 5.1. *Let P_1 and P_2 be two atomsets at the same level l, defined as in Definition 5.1. If $P_1 \succeq_\theta P_2$, then $sup(P_1, D) \geq sup(P_2, D)$.*

It is noteworthy that if $P_1 \succeq_\theta P_2$ and P_1 is not frequent ($sup(P_1, D) < \sigma_{min}[l]$), then also P_2 is not frequent ($sup(P_2, D) < \sigma_{min}[l]$). This monotonicity property of $\succeq_\theta$ with respect to the support allows for pruning the search space without losing frequent atomsets.

In the inter-level search, atomsets discovered at level l are refined by descending the generalization hierarchies up to finding task-relevant objects mapped at level $l + 1$. These are the only candidate atomsets considered for evaluation, since other candidates would not meet the necessary condition for atomsets to be frequent at level $l + 1$ when $\sigma_{min}[l + 1] \leq \sigma_{min}[l]$ (see Definition 5.4). This way, the search space at level $l + 1$ is heavily pruned. Moreover, information on the units of analysis covered by atomsets at level l can be used to make more efficient the evaluation of the support of atomsets at level $l + 1$. Indeed, if a unit of analysis $D[s]$ is not covered by a pattern P at granularity level l, then it will not be covered by any descendant of P at level $l + 1$.

Once frequent atomsets have been generated at level l, it is possible to generate *strong* spatial association rules, i.e., rules whose confidence is higher than a threshold $\kappa_{min}[l]$. In particular, each frequent atomset P at level l is partitioned into two atomsets A and C such that $P = A \wedge C$ and the confidence of the association rule $A \Rightarrow C$ is computed. Different partitions of P generate different association rules. Those association rules with confidence lower than $\kappa_{min}[l]$ are filtered out.

We conclude by observing that in real-world applications a large number of frequent atomsets and strong association rules can be generated, most of which are uninteresting. This is also true for the module discovery problem (e.g., constituent motifs with a large inter-motif distance). To prevent this, some pattern constraints can be expressed in a declarative form and then used to filter out uninteresting atomsets or spatial association rules [4].

5.4 Implementation

The development of a module discovery tool effectively usable by biologists demands for the solution of several problems, both methodological and architectural. Methodological problems involve data pre-processing, namely discretization of numerical data, and the automated selection of some critical parameters such as minimum support. Architectural problems concern the interface of the tool with the external world, either to acquire data and parameters or to communicate results. In this section, solutions to these problems are briefly reported.

5.4.1 Choosing the Minimum Support Threshold

Setting up the minimum support threshold σ_{min} is not a trivial problem for a biologist when assuming no a priori knowledge about structural and functional features

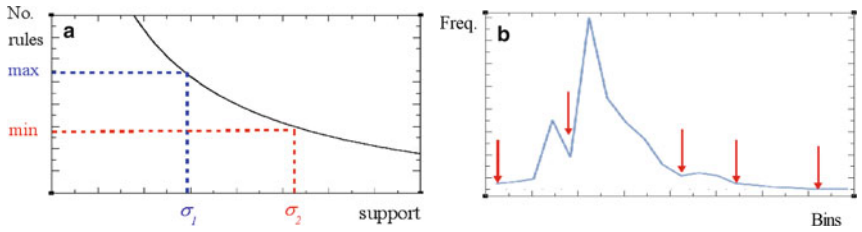

Fig. 5.3 (**a**) Functional dependence of the number of spatial association rules from the minimum support threshold. (**b**) Histogram and minimum points

Algorithm 1 Automated setting of σ_{min}

```
1: find_minsup(i,[σ1, σ2],[min, max])
2: if i ≥ MAX_ITERS then
3:    return (σ1 + σ2)/2
4: end if
5: no_Rules ← SPADA((σ1 + σ2)/2)
6: if no_Rules ≥ max then
7:    σmin ← find_minsup(i + 1, [σ1, (σ1 + σ2)/2],[min, max])
8: else if no_Rules ≤ min then
9:    σmin ← find_minsup(i + 1, [(σ1 + σ2)/2, σ2],[min, max])
10: else
11:    σmin ← (σ1 + σ2/2)
12: end if
13: return σmin
```

of potential modules. For this reason, we follow the approach suggested in [23]: users are asked to choose an interval $[min, max]$ for the number of association rules they want to examine, and a value for σ_{min} is then automatically derived. Indeed, the number of association rules generated by SPADA depends on σ_{min} according to some function ϕ, which is monotonically decreasing (Fig. 5.3a). Therefore, the selection of an interval $[min, max]$ for the number of association rules corresponds to the selection of an interval $[\sigma_1, \sigma_2]$ for the support, which includes the optimal value σ_{min}.

Contrary to [23], where a linear search of the optimal value is proposed, we apply a dichotomic search for efficiency reasons. The formulation of the algorithm is recursive (see Algorithm 1). Initially, the procedure *find_minsup* is invoked on the support interval $[0, 1]$ and SPADA is run with $\sigma_{min} = 0.5$. If necessary, *find_minsup* is recursively invoked on either $[0, 0.5]$ or $[0.5, 1]$. Since the convergence of the algorithm cannot be proven, we stop the search when the number of recursive invocations exceeds a maximum iteration threshold *MAX_ITERS*. A reasonable setting is $MAX\_ITERS = 5$, since after five iterations, the width of the interval $[\sigma_1, \sigma_2]$ is relatively small ($\frac{1}{2^5}$).

5.4.2 Discretizing Numerical Values

SPADA cannot properly deal with numerical values of inter-motif distances. Therefore, it is necessary to transform them into categorical values through some discretization technique. The equal frequency (EF) discretization algorithm partitions the initial range of values into a fixed number of intervals (or *bins*), such that they have different width but approximately the same number of values. This partitioning may significantly affect the subsequent rule mining step, but unfortunately, choosing a suitable number of bins is by no means an easy task for a biologist. For this reason, we investigated a new algorithm which, similarly to EF, partitions the initial range according to data distribution, but, differently from EF, it needs no input parameter.

This algorithm, called DUDA (**D**ensity-based **U**nsupervised **D**iscretization **A**lgorithm), is mainly inspired by clustering algorithms based on kernel density estimation [17], which groups together data that follow the same (typically normal) distribution (see Algorithm 2). Histograms are used to model the distribution of numerical data (inter-motif distances). The width w of each bin is computed by resorting to Scott's formula [40] $w = \frac{3.5 \times s}{\sqrt[3]{n}}$, where n is the number of values to discretize and s is the standard deviation of the values.

In this work, we look for bins so that the values in each bin are normally distributed. Partitions are identified by finding relative minimums in the histogram of frequency distribution (Fig. 5.3b), which are candidate split points for the partitioning.

Once the initial partitioning is defined, the algorithm works iteratively: at each iteration, it tries to merge two consecutive bins. Merging is performed when the distribution of values in the partition obtained after merging fits a normal distribution better than the two original bins. The decision of merging is based on the Kolmogorov–Smirnov normality test, which typically works by verifying the null

Algorithm 2 DUDA: Density-based Discretization Algorithm

```
1:  DUDA(P,F)
2:  if number_of_partitions(P) > 1 then
3:     bestL ← 0
4:     for (a,b) ∈ get_consecutive_partitions(P) do
5:        if L_{a,b} < bestL then
6:           (best_a, best_b) ← (a,b)
7:           bestL ← L_{a,b}
8:        end if
9:     end for
10:    if bestL < 0 then
11:       return DUDA(merge(best_a, best_b, P), mergeF(best_a, best_b, F))
12:    end if
13: end if
14: return P
```

hypothesis "H_0: data are normally distributed," given a confidence level. In our case, we find the minimum confidence level α for which H_0 is not rejected, and we use it to identify the best merging according to the following formula:

$$L_{a,b} = \alpha_{a,b} \cdot (F_a + F_b) - (\alpha_a \cdot F_a + \alpha_b \cdot F_b), \tag{5.7}$$

where:

- F_a (F_b) is the relative frequency of values in the partition a (b)
- α_a, α_b and $\alpha_{a,b}$ are the confidence values of the Kolmogorov–Smirnov test on a, b and on the partition obtained after merging a and b, respectively;
- $L_{a,b}$ is the *loss* obtained after merging a and b.

Obviously, the smaller $L_{a,b}$, the better. The iteration stops when all possible $L_{a,b}$ are positive (no improvement is obtained) or no further merging is possible. The algorithm is recursive: it takes as input the list of partitions and the list of frequencies and returns a new list of partitions. The functions *merge* and *mergeF* take as input a list of r elements and return a list of $r-1$ elements, where two consecutive elements are appropriately merged.

5.4.3 Data Acquisition and Result Processing

SPADA has been integrated in a system which takes the set T of sequences from a text file. This file is processed in order to mine frequent motif sets as presented in Sect. 5.2.1. The output of this first step is an XML file which is stored in an XML repository. The corresponding document type definition (DTD) is shown in Fig. 5.4. For each frequent motif set S ($|t_S| \geq \tau_{min}$), the XML file describes the support set t_S together with some simple statistics (e.g., the ratio $|t_S|/|T|$). The module that implements the discretization algorithm DUDA (see Sect. 5.4.2) operates on data stored in the XML repository.

A wrapper of SPADA loads XML data in the extensional part D_E of the deductive database D used by SPADA itself, while rules of the intensional part D_I can be

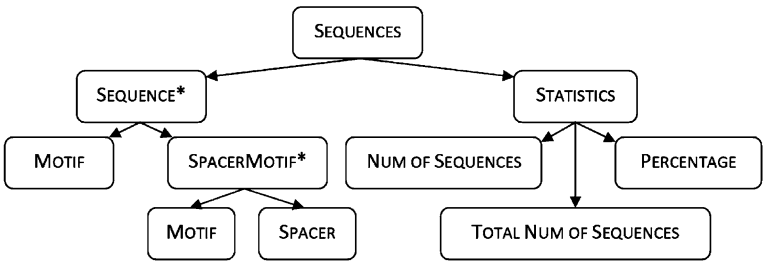

Fig. 5.4 Hierarchical structure arrangement of elements of the XML document type definition

edited by the user through a graphical user interface. This wrapper is also in charge of automatically setting the σ_{min} parameter as per Algorithm 1 in Sect. 5.4.1.

By merging consecutive bins through the rules in D_I, many spatial association rules are discovered, which differ only in some intervals of inter-motif distances. An unmanageably large number of association rules makes interpretation of results cumbersome for the biologist. For this reason, association rules are filtered before being shown to the user. Three filtering criteria are considered. The first criterion selects the association rules with the smallest bins among rules with the same motifs, the same confidence and supported by the same sequences. The second criterion selects the association rules with the greatest support among those with the same motifs and confidence, whose bins are included in the bins of the selected rules and whose list of supporting sequences is included in the list of supporting sequences of the selected rules. The last criterion selects the association rules with highest confidence among those with the same motifs, whose bins are included in the bins of the selected rules and whose list of supporting sequences is included in the list of supporting sequences of the selected rules.

5.5 Case Study

To show the potential of the integrated system, a pilot study is conducted on translation regulatory motifs located in the nucleotide sequences of untranslated regions (UTRs) of nuclear transcripts (mRNAs) targeting mitochondria. These motifs are essential for mRNA subcellular localization, stability and translation efficiency [50]. Evidence from recent studies supports the idea that the nature and distribution of these translation regulatory motifs may play an important role in the differential expression of mRNAs [11].

Datasets are generated as a view on three public biological databases, namely MitoRes,[6] UTRef and UTRsite.[7] The view integrates data on UTR sequences and their contained motifs, together with information on the motifs width and their starting and ending position along the UTR sequences in the UTRminer [48] database. We base our analysis on a set T of 728 3′UTR sequences relative to the human species. Twelve motifs are initially considered (set M). By setting $\tau_{min} = 4$, several frequent motif sets (set $\mathcal{S}$) are extracted in the first phase. We focus our attention on the motif set $S \in \mathcal{S}$ with the largest support set (111 3′UTR sequences). It contains three motifs, which are denoted as x, y and z. The hierarchy defined on motifs has three levels (Fig. 5.2), but we consider only the middle level, since the top level conveys little information on the constituent motifs of a module, while the bottom level is too specific to find interesting rules.

[6] http://www2.ba.itb.cnr.it/MitoRes/

[7] http://utrdb.ba.itb.cnr.it/

To discretize inter-motif distances, both EF and DUDA discretization are tested with two settings of the threshold σ_{min} (40% and 50%) and one of κ_{min} (80%). The number of intervals set for EF is 12. Since we have no prior knowledge on the suitability of this choice, we intentionally define some distance predicates whose semantics correspond to a merging operation of consecutive intervals (rules (5.3) reported in Sect. 5.2.2 exemplify intensionally defined distance predicates for intervals merging). This way, the comparison between EF and the discretization method proposed in this chapter is fair and does not depend on our initial decision of partitioning the distances in 12 intervals.

Experimentally, we observe that the running time varies significantly between the two solutions (Table 5.1). Indeed, the use of intensionally defined predicates to merge intervals slows down the discovery process and has the undesirable effect of returning a large number of similar rules which have to be finally filtered out.

We also test the procedure for the automated selection of the σ_{min} threshold. The interval chosen for the number of spatial association rules is $[min = 50, max = 100]$, while $MAX\_ITERS = 6$. After five steps, the system converges to $\sigma_{min} = 0.5313$ and returns 85 spatial association rules (Table 5.2).

An example of spatial association rule discovered by SPADA is the following:

$$\begin{aligned} &sequence(T), part\_of(T, M_1), is\_a(M_1, x), distance(M_1, M_2, [-99.. -18]), \\ &is\_a(M_2, y), distance(M_2, M_3, [-99..3.5]), M_1 \neq M_2, M_1 \neq M_3, M_2 \neq M_3 \\ &\Rightarrow is\_a(M_3, z) \end{aligned} \tag{5.8}$$

This rule can be interpreted as follows: if a motif of type x is followed by a motif of type y, their inter-motif distance falls in the interval $[-99.. -18]$, and the motif of type y is followed by another motif at an inter-motif distance which falls in the interval $[-99..3.5]$, then that motif is of type z. The support of this rule is 63.96%, while the confidence is 100%. The high support reveals a statistically overrepresented module, which may be indicative of an associated biological function. This module can also be represented by the following chain:

Table 5.1 Results for the two discretization algorithms

	σ_{min}	Running time	No. of unfiltered rules	No. of filtered rules
Equal frequency	40%	>36 h	1,817	84
	50%	>4 h	220	36
DUDA	40%	4 s	16	16
	50%	1 s	12	12

Table 5.2 Choosing the minimum support threshold

Iteration no.	1	2	3	4	5
No. rules	185	9	25	40	85
σ_r	0.5000	0.7500	0.6250	0.5625	0.5313

$$\langle x, [-99..-18], y, [-99..3.5], z \rangle,$$

which is similar to that reported as (5.1) in Sect. 5.2.2, with the difference that here intervals of inter-motif distances are reported. The high confidence means that when the conditions expressed in the antecedent hold, the type z of the third motif in the chain can be predicted with certainty. Therefore, the spatial association rule conveys additional inferential information with respect to the frequent pattern.

5.6 Conclusions

In this chapter, we describe a new approach to module discovery by mining spatial association rules from a set of biological sequences where type and position of regulatory single motifs are annotated. The method is based on an ILP formulation which facilitates the representation of the biological sequences by means of sets of Datalog ground atoms, the specification of background knowledge by means of Datalog rules and the formulation of pattern constraints by means of a declarative formalism. Although results of the method are easy to read for a data-mining expert, they are not intelligible for a biologist because of the use of first-order logic to represent spatial patterns. For this reason, the spatial association rule mining method has been implemented in a tool which effectively support biologists in module discovery tasks by graphically rendering mined association rules. The tool also supports biologists in other critical decisions, such as selecting the minimum support threshold. To face the hard discretion problem, which typically affects discrete approaches like that described in this chapter, we have also implemented a new discretization method, which is inspired by kernel density estimation-based clustering and needs no input parameters.

The tool has been applied to a pilot study on translation regulatory motifs located in the untranslated regions of messenger RNAs targeting mitochondria. The application shows the potential of the approach and methods proposed in this chapter.

Acknowledgements This work is supported in partial fulfillment of the research objectives of two Italian projects funded by the MIUR (Italian Ministry for Education, University and Research): "LIBi: Laboratorio Internazionale di Bioinformatica" (FIRB project) and "MBLab: Laboratorio di Bioinformatica per la Biodiversità Molecolare" (FAR project). The authors gratefully acknowledge Lynn Rudd for reading the initial version of this chapter.

References

1. Aerts, S., Loo, P.V., Thijs, G., Moreau, Y., Moor, B.D.: Computational detection of cis-regulatory modules. In: Proc. of the European Conf. on Computational Biology (ECCB), pp. 5–14 (2003)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. of the 21st Int. Conf. on Very Large Data Bases, pp. 487–499 (1994)

3. Agrawal, R., Srikant, R.: Mining sequential patterns. In: P.S. Yu, A.L.P. Chen (eds.) Proc. of the 11th Int. Conf. on Data Engineering (ICDE), pp. 3–14. IEEE Computer Society (1995)
4. Appice, A., Berardi, M., Ceci, M., Malerba, D.: Mining and filtering multi-level spatial association rules with ares. In: M.S. Hacid, N.V. Murray, Z.W. Ras, S. Tsumoto (eds.) Foundations of Intelligent Systems, 15th Int. Symposium, ISMIS 2005, *LNCS*, vol. 3488, pp. 342–353. Springer (2005)
5. Bailey, T.L., Elkan, C.: Fitting a mixture model by expectation maximization to discover motifs in biopolymer. In: R.B. Altman, D.L. Brutlag, P.D. Karp, R.H. Lathrop, D.B. Searls (eds.) Proc. of the 2nd Int. Conf. on Intelligent Systems for Molecular Biology (ISMB), pp. 28–36. AAAI (1994)
6. Bi, C.: Seam: a stochastic EM-type algorithm for motif-finding in biopolymer sequences. Journal of Bioinformatics and Computational Biology **5**(1), 47–77 (2007)
7. Blockeel, H., Sebag, M.: Scalability and efficiency in multi-relational data mining. SIGKDD Explorations **5**(1), 17–30 (2003)
8. Buhler, J., Tompa, M.: Finding motifs using random projections. Journal of Computational Biology **9**(2), 225–242 (2002)
9. Ceri, S., Gottlob, G., Tanca, L.: Logic programming and databases. Springer, New York (1990)
10. Dehaspe, L., De Raedt, L.: Mining association rules in multiple relations. In: the 7th Int. Workshop on Inductive Logic Programming, ILP 1997, vol. 1297, pp. 125–132. Springer (1997)
11. Didiano, D., Hobert, O.: Molecular architecture of a miRNA-regulated 3'UTR. RNA (New York) **14**(7), 1297–1317 (2008)
12. Erman, B., Cortes, M., Nikolajczyk, B., Speck, N., Sen, R.: Ets-core binding factor: a common composite motif in antigen receptor gene enhancers. Molecular and Cellular Biology **18**(3), 1322–1330 (1998)
13. Frith, M.C., Hansen, U., Weng, Z.: Detection of cis-element clusters in higher eukaryotic DNA. Bioinformatics **17**(10), 878–889 (2001)
14. Gupta, M., Liu, J.S.: De novo cis-regulatory module elicitation for eukaryotic genomes. Proc. National Acadademy of Science **102**(20), 7079–7084 (2005)
15. Heinemeyer, T., Wingender, E., Reuter, I., Hermjakob, H., Kel, A.E., Kel-Margoulis, O.V., Ignatieva, E.V., Ananko, E.A., Podkolodnaya, O.A., Kolpakov, F.A., Podkolodny, N.L., Kolchanov, N.A.: Databases on transcriptional regulation: TRANSFAC, TRRD and COMPEL. Nucleic Acids Research **26**(1), 362–367 (1998)
16. Helft, N.: Inductive generalization: a logical framework. In: I. Bratko, N. Lavrač (eds.) Progress in Machine Learning, pp. 149–157. Sigma Press, Wilmslow (1987)
17. Hinneburg, A., Keim, D.A.: A general approach to clustering in large databases with noise. Knowledge and Information Systems **5**(4), 387–415 (2003)
18. Ivan, A., Halfon, M., Sinha, S.: Computational discovery of cis-regulatory modules in drosophila without prior knowledge of motifs. Genome Biology **9**(1), R22 (2008)
19. Jackups, R., Liang, J.: Combinatorial analysis for sequence and spatial motif discovery in short sequence fragments. IEEE/ACM Trans. Comput. Biology Bioinform. **7**(3), 524–536 (2010)
20. Johansson, Ö., Alkema, W., Wasserman, W.W., Lagergren, J.: Identification of functional clusters of transcription factor binding motifs in genome sequences: the mscan algorithm. Bioinformatics **19 (suppl 1)**, i169–i176 (2003)
21. Klepper, K., Sandve, G.K., Abul, O., Johansen, J., Drabløs, F.: Assessment of composite motif discovery methods. BMC Bioinformatics **9**, 123 (2008)
22. Li, M., Ma, B., Wang, L.: Finding similar regions in many sequences. Journal of Computer and System Sciences **65**(1), 73–96 (2002)
23. Lin, W., Alvarez, S.A., Ruiz, C.: Efficient adaptive-support association rule mining for recommender systems. Data Mining and Knowledge Discovery **6**(1), 83–105 (2002)
24. Lisi, F.A., Malerba, D.: Inducing multi-level association rules from multiple relations. Machine Learning **55**(2), 175–210 (2004)
25. Liu, X., Brutlag, D.L., Liu, J.S.: Bioprospector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. In: Pacific Symposium on Biocomputing, pp. 127–138 (2001)

26. MacIsaac, K.D., Fraenkel, E.: Practical strategies for discovering regulatory DNA sequence motifs. PLoS Compututational Biology **2**(4), e36 (2006)
27. Malerba, D., Lisi, F.A.: An ILP method for spatial association rule mining. In: In Working notes of the First Workshop on Multi-Relational Data Mining, pp. 18–29 (2001)
28. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery **1**(3), 241–258 (1997)
29. Mitchell, T.: Machine Learning. McGraw-Hill, NY (1997)
30. Muggleton, S., Srinivasan, A., King, R.D., Sternberg, M.J.E.: Biochemical knowledge discovery using inductive logic programming. In: S. Arikawa, H. Motoda (eds.) Discovery Science, *LNCS*, vol. 1532, pp. 326–341. Springer, Berlin (1998)
31. Nienhuys-Cheng, S.H., De Wolf, R.: Foundations of Inductive Logic Programming, *LNAI*, vol. 1228. Springer, Berlin (1997)
32. Perdikuri, K., Tsakalidis, A.K.: Motif extraction from biological sequences: Trends and contributions to other scientific fields. In: Proc. of the 3rd Int. Conf on Information Technology and Applications (ICITA), vol. 1, pp. 453–458. IEEE Computer Society (2005)
33. Plotkin, G.D.: A note on inductive generalization. Machine Intelligence **5**, 153–163 (1970)
34. Remnyi, A., Schler, H.R., Wilmanns, M.: Combinatorial control of gene expression. Nature Structural & Molecular Biology **11**(9), 812–815 (2004)
35. Rigoutsos, I., Floratos, A.: Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm [published erratum appears in bioinformatics 1998;14(2): 229]. Bioinformatics **14**(1), 55–67 (1998)
36. Robin, S., Rodolphe, F., Schbath, S.: DNA, Words and Models: Statistics of Exceptional Words. Cambridge University Press, London (2005)
37. Sandelin, A., Alkema, W., Engström, P.G., Wasserman, W.W., Lenhard, B.: JASPAR: an open-access database for eukaryotic transcription factor binding profiles. Nucleic Acids Research **32**(Database-Issue), 91–94 (2004)
38. Sandve, G.K., Drabløs, F.: Generalized composite motif discovery. In: R. Khosla, R.J. Howlett, L.C. Jain (eds.) Knowledge-Based Intelligent Information and Engineering Systems, 9th Int. Conf., KES 2005, vol. 3, *LNCS*, vol. 3683, pp. 763–769. Springer (2005)
39. Sandve, G.K., Abul, O., Drabløs, F.: Compo: composite motif discovery using discrete models. BMC Bioinformatics **9** (2008)
40. Scott, D.: On optimal and data-based histograms. Biometrika **66**, 605–610 (1979)
41. Segal, E., Sharan, R.: A discriminative model for identifying spatial cis-regulatory modules. Journal of Computational Biology **12**(6), 822–834 (2005)
42. Sharan, R., Ovcharenko, I., Ben-Hur, A., Karp, R.M.: CREME: a framework for identifying cis-regulatory modules in human-mouse conserved segments. Bioinformatics **19 (suppl 1)**(18), S283–S291 (2003)
43. Sinha, S., Tompa, M.: A statistical method for finding transcription factor binding sites. In: P.E. Bourne, M. Gribskov, R.B. Altman, N. Jensen, D.A. Hope, T. Lengauer, J.C. Mitchell, E.D. Scheeff, C. Smith, S. Strande, H. Weissig (eds.) ISMB, pp. 344–354. AAAI (2000)
44. Srinivasan, A., King, R.D., Muggleton, S., Sternberg, M.J.E.: Carcinogenesis predictions using ILP. In: N. Lavrac, S. Dzeroski (eds.) Inductive Logic Programming, 7th International Workshop, ILP-97, *LNCS*, vol. 1297, pp. 273–287. Springer (1997)
45. Srinivasan, A., King, R.D., Muggleton, S., Sternberg, M.J.E.: The predictive toxicology evaluation challenge. In: Proc. of the 15th Int. Joint Conf. on Artificial Intelligence (IJCAI), pp. 4–9 (1997)
46. Stormo, G.D.: DNA binding sites: representation and discovery. Bioinformatics **16**(1), 16–23 (2000)
47. Takusagawa, K.T., Gifford, D.K.: Negative information for motif discovery. In: R.B. Altman, A.K. Dunker, L. Hunter, T.A. Jung, T.E. Klein (eds.) Pacific Symposium on Biocomputing, pp. 360–371. World Scientific, Singapore (2004)
48. Turi, A., Loglisci, C., Salvemini, E., Grillo, G., Malerba, D., D'Elia, D.: Computational annotation of UTR cis-regulatory modules through frequent pattern mining. BMC Bioinformatics **10 (suppl 6)**, S25 (2009)

49. Valiant, L.G.: A theory of the learnable. Communications of the ACM **27**(11), 1134–1142 (1984)
50. Wilkie, G., Dickson, K., Gray, N.: Regulation of mRNA translation by 5'- and 3'-UTR-binding factors. Trends in Biochemical Sciences **28**(4), 182–188 (2003)
51. Xing, E.P., Wu, W., Jordan, M.I., Karp, R.M.: Logos: a modular bayesian model for de novo motif detection. Journal of Bioinformatics and Computational Biology **2**(1), 127–154 (2004)
52. Zhou, Q., Wong, W.H.: CisModule: De novo discovery of cis-regulatory modules by hierarchical mixture modeling. Proceedings of the National Academy of Sciences of the United States of America **101**(33), 12114–12119 (2004)