Discovering *Cis*-Regulatory Modules by Optimizing Barbecues

Axel Mosig
Türker Biyikoglu
Sonja J. Prohaska
Peter F. Stadler

SFI WORKING PAPER: 2007-08-025

SANTA FE INSTITUTE

Discovering *Cis*-Regulatory Modules by Optimizing Barbecues

Axel Mosig [a,b], Türker Bıyıkoğlu [c], Sonja J. Prohaska [d,e], Peter F. Stadler [e,g,f,h]

[a] *CAS-MPG Partner Institute for Computational Biology, Shanghai Institutes for Biological Sciences, 320 Yue Yang Road, 200031 Shanghai, China*
axel.mosig@gmail.com

[b] *Max-Planck Institute for Mathematics in the Sciences, Inselstrasse 22, D-04103 Leipzig, Germany*

[c] *Işık University, Kumbaba Mevkii Şile, 34980, Turkey*
turker.biyikoglu@isikun.edu.tr

[d] *Department of Biomedical Informatics, School of Computing and Informatics, Arizona State University, Tempe, PO-Box 878809, AZ 85287, USA*
sonja.prohaska@asu.edu

[e] *Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center for Bioinformatics, University of Leipzig, Härtelstr. 16-18, D-04107 Leipzig, Germany*
studla@bioinf.uni-leipzig.de

[f] *Department of Theoretical Chemistry, University of Vienna, Währingerstraße 17, A-1090 Wien, Austria*

[g] *Fraunhofer Institut für Zelltherapie und Immunologie, Deutscher Platz 5e, D-04103 Leipzig, Germany*

[h] *Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA*

Abstract

Gene expression in eukaryotic cells is regulated by a complex network of interactions, in which transcription factors and their binding sites on the genomic DNA play a determining role. As transcriptions factor rarely, if ever, act in isolation, binding sites of interacting factors are typically arranged in close proximity forming so-called *cis*-regulatory modules. Even when the individual binding sites are known, module discovery remains a hard combinatorial problem, which we formalize here as the *Best Barbecue Problem*. It asks for simultaneously stabbing a maximum number of differently colored intervals from K arrangements of colored intervals. This geometric problem turns out to be an elementary, yet previously unstudied combinatorial optimization problem of detecting common edges in a family of hypergraphs, a decision version of which we show here to be NP-complete. Due to its relevance in biological applications, we propose algorithmic variations that are suitable for the

analysis of real data sets comprising either many sequences or many binding sites. Being based on set systems induced by interval arrangements, our problem setting generalizes to discovering patterns of co-localized itemsets in non-sequential objects that consist of corresponding arrangements or induce set systems of co-localized items. In fact, our optimization problem is a generalization of the popular concept of *frequent itemset mining*.

Key words: gene regulation, *cis*-regulatory modules (CRMs), Best Barbecue Problem, NP-completeness, branch-and-bound algorithms, itemset mining

---

1 Introduction and Biological Background

A comprehensive understanding of the mechanism of eukaryotic gene expression is a major challenge in current research in molecular biology. The regulation of transcription by means of DNA-binding transcription factors forms a key component of gene regulation networks. In general, the binding of multiple transcription factors in specific combinations is required for proper regulation. The corresponding transcription factor binding sites (TFBS) on the DNA sequence thus form so-called *cis*-regulatory modules (CRMs) [24]. From a biological point of view, CRMs are defined as independent DNA elements that exert specific regulatory functions on a nearby gene due the binding of activating and/or repressing transcription factors [3]. From a hands-on computational biology point of view, CRMs are DNA sequences of limited length (in the range of a few hundred nucleotides) that contain a number of short DNA motifs which correspond to binding sites of individual transcription factors [6,12,14,36,37]. The functional importance of CRMs is highlighted for instance by the observation that a significant fraction of the tissue-specific gene expression can be explained by a limited number of CRMs in the proximal promoters of mammalian genes [33].

Although there are extensive databases of individual transcription factors and their corresponding binding sites [15,28], it is still a hard problem to distinguish *bona fide* CRMs from spurious combinations of TFBSs [7]. A major complication is the fact that the DNA patterns bound by most transcription factors are very short and promiscuous [35]. As a consequence, predicted TFBSs cover the genome almost completely. This makes the computational discovery of CRMs from genomic sequence data a challenging task. Typically, this issue is approached by comparing the promoter regions (which range up to several thousand nucleotides upstream or in some cases even downstream of the transcription start site) flanking the coding regions of sets of genes. Typically, one either considers the promoter regions of evolutionarily related genes *across different species* (so-called "phylogenetic footprinting"), or one

attempts to detect common sequence motifs in genes with similar expression profiles *within one species*. In recent studies, the two approaches are often combined to increase the specificity of the procedures, e.g. [18]. Although some insight has been gained on conservation and loss of regulatory sequences, the mechanisms underlying their evolution still remain largely enigmatic. While sequence conservation is a suitable indicator of conserved regulatory function, the absence of sequence conservation does not imply loss of regulatory function [34,11,13,22]. This phenomenon has first been documented in the Drosophila *even skipped stripe 2 enhancer* [19], and is usually referred to as binding-site turnover; we will return to this point in Section 6.1.

Recent observations [10,29] indicate explicitly that shuffling of the relative positions of conserved elements is a major mode of evolution for *cis*-regulatory elements. In other words, CRMs conserve their types of TFBSs but not necessarily their order along the genomic DNA sequence. Note that due to these shuffling effects, traditional edit-distance-based alignment procedures are not appropriate tools for unveiling regulatory modules. Novel approaches are thus required for such "non-order-preserving" alignments.

The *Best Barbecue* (BBQ) approach explored in this contribution is based on discovering sets of binding sites that occur close to each other in several promoter sequences, where the notion of "close to each other" is made precise by requiring that the TFBSs occur within an interval of fixed length. In [8], the problem of CRM discovery is described in a way that is similar to our approach. Instead of attempting a provably optimal solution, a heuristic algorithm is used for module discovery, however. Genetic Algorithms are used in [1] and [25]. `CREME` [30], which is also conceptually related to our BBQ approach, is probably the most widely used method. This program seeks to identify motif clusters of limited length that occur more than once in a set of genomic sequences. In contrast to BBQ, the modules discovered by `CREME` contain *precisely* the same set of binding sites and may not contain additional binding sites. This restriction is not realistic for both biological and methodological reasons. It is plausible that a functional regulatory module may contain a putative binding site for a transcription factor that is not involved in the module's function: the additional transcription factor could be down-regulated while the regulatory module is active, relative locations of the binding sites might not allow the additional factor to become part of the protein complex, and binding site profiles with low sequence specificity frequently produce false positive matches.

Several other approaches to discovering regulatory modules have been investigated. Kel-Margoulis *et al.* [17] propose a method based on identifying clusters with the property that pairwise distances between occurrences of TFBSs fall within certain bounds; sets of binding sites that maximize a certain cluster score are searched by the means of a genetic algorithm. Recently, Schones *et al.*

studied the statistics of binding site co-occurrences to obtain probabilities for observing regulatory modules that satisfy different constraints regarding either the order and orientation of binding sites or the gaps in-between them. Noto et al. [23] train HMMs that reflect certain logical and spatial relationships between the binding sites of the regulatory modules to be detected. Other methods are based on probabilistic methods [26] or require (only sparsely available) knowledge about interactions between transcription factors such as the algorithm presented in [32].

Given the practical importance of CRM discovery it seems natural to raise the question how complex regulatory module discovery is. A major goal of our contribution is therefore to put the increasingly important task of regulatory module discovery on a formal basis, provide insights into issues which make the problem difficult, and suggest how algorithms can be devised that yield provably optimal results under certain relaxed problem specifications. We show that our abstract and very general way of looking at regulatory module discovery leads to a natural combinatorial and geometric optimization problem that is NP-complete in general. As a practical variant, we propose a slightly modified problem that can be solved with algorithms whose time complexity is exponential in the maximum number of binding sites that are not shared among the regulatory modules to be discovered. Furthermore, our approach can be equipped with different scoring schemes, which are relevant for practical use. As an example, we demonstrate the feasibility of the BBQ approach on intergenic regions of *Hox* gene clusters.

The outline of this paper is as follows: we start with a formal description of regulatory module discovery; although our starting point is a string matching problem, it turns out that taking a geometric point of view is much more convenient in this setting. Our geometric characterization leads to the *Best Barbecue Problem*, which, to the best of the authors' knowledge, has not been studied previously. The Best Barbecue Problem deals with simultaneously stabbing intervals of the same color from several interval arrangements and can be rephrased as a combinatorial optimization problem. In Section 4.1, we show that the Best Barbecue Problem and its variants are NP-complete. We then provide branch-and-bound-like algorithms, with some results from a biological application demonstrating the practical relevance of the problem. Furthermore, we provide an algorithm that is exponential in an additional input parameter that can be assumed to be small in practice, but yields correct solutions only for certain (well characterized) instances. Each of the algorithms we present is exponential in a different input parameter, hence the algorithms are useful for different types of instances. As a final contribution, we show that a slight extension of our problem setting leads to a natural generalization of the well known concept of frequent itemset mining.

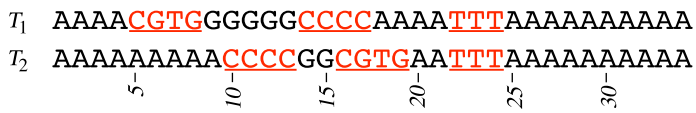

Fig. 1. Examples of an L-occurrence of $S = \{s_1, s_2, s_3\}$ with $s_1 = \texttt{CCCC}$, $s_2 = \texttt{CGTG}$, and $s_3 = \texttt{TTT}$ in the sequences T_1 and T_2, respectively. For T_1, we have an L-occurrence for any $L \geq 20$, in case of T_2 for $L \geq 15$.

2 L-occurrences and Interval Arrangements

Throughout this paper, let Σ denote some finite alphabet. When dealing with genome sequences, we usually have $\Sigma = \{A, C, G, T\}$ denoting the four types of nucleotides occurring in DNA. As a notational convention, let $[a : b] := \{a, a+1, \ldots, b\}$ denote the integer interval from a to b for any two integers a, b if $a \leq b$. Given an integer μ and an integer interval $[a : b]$, we say that μ *stabs* $[a : b]$ iff $\mu \in [a : b]$. Furthermore, given a string $T = \tau_1 \ldots \tau_n$, let $|T|$ denote its length, and for any two integers a, b we write $T|_{a,b}$ for the substring $\tau_a \tau_{a+1} \ldots \tau_b$. We say that a string U *occurs in* T *at position* x iff $1 \leq x \leq x + |U| - 1 \leq n$ and $T|_{x,x+|U|-1} = U$. Due to the combinatorial nature of our original problem, all our considerations will refer to integer intervals. Many results that we obtain, however, hold for intervals over the reals as well.

As mentioned above, *cis*-regulatory modules are *clustered* occurrences of TFBSs along a genome. We formally grasp the notion of clustered occurrences, Fig. 1, by introducing a cluster length L and say that binding site occurrences are L-clustered if the occurrences are contained within an interval of size L along the genome:

Definition 1 *Let $S = \{s_1, \ldots, s_m\} \subseteq \Sigma^*, T \in \Sigma^*$, $L \in \mathbb{N}$ and $A \subseteq S$. We say that A* is an L-occurrence in T w.r.t. S *if there is a mapping $x\colon A \to \mathbb{N}$ associating an index x_s (indicating a position in T where s occurs) with each $s \in A$ such that*

(O1) *s occurs in T at position x_s for each $s \in A$ and*
(O2) $\max(x_s + |s|, x_t + |t|) - \min(x_s, x_t) \leq L$ *for all $s, t \in A$.*

Correspondingly, we refer to A together with the mapping x satisfying the above conditions as an L-occurrence *of A in T w.r.t. S.*

Note that in the above definition, the *complete* sequences in A – not just their starting positions – occur within a range of L nucleotides in T. In the case of two sequences, L-occurrences are somewhat related to gene teams [5], the two differences being that (a) the occurrences of "binding sites" are rather positions of genes on chromosomes and (b) gene teams require constraints on

distances between *each consecutively occurring* pair of genes s and t rather than between *all* pairs of genes. In the first simplistic scenario to be considered here, we are interested in finding L-occurrences of maximum cardinality. Moreover, we are interested in finding L-occurrences that can be observed simultaneously in several sequences $T_1, \ldots, T_K$. This leads to the following optimization and decision problems:

Problem 1 Maximum Simultaneous L-Occurrence (MSLO)
INSTANCE: Integer L; $T_1, \ldots, T_K$ and $s_1, \ldots, s_m$ denoting strings over an alphabet Σ.
TASK: Determine the maximum cardinality of a set $A \subseteq \{s_1, \ldots, s_m\}$ such that A is an L-occurrence w.r.t. $\{s_1, \ldots, s_m\}$ in each of the sequences $T_1, \ldots, T_K$.

Later on, we will be particularly interested in the decision version of the problem: Rephrased as a decision problem, we are given an additional threshold parameter θ and ask whether the maximum cardinality simultaneous L-occurrence exceeds θ. We will refer to the decision version as *DSLO*.

For the biological application of regulatory module discovery, we are interested in the "most surprising" rather than the largest cardinality L-occurrence. This is, in fact, achieved through weighting schemes discussed in Section 6.3. For the sake of clarity, however, the following considerations on algorithms and complexity refer to the unweighted scenario.

Before dealing with the complexity of MSLO and DSLO, we step back and study the scenario involving a single sequence T in more detail. A building block of the algorithms we develop in the sequel is a certain set of colored intervals. We write colored intervals as pairs, i.e., $([h : i], c)$ denotes the interval $[h : i]$ with color $c \in [1 : m]$. Given $S = \{s_1, \ldots, s_m\}$ as in Definition 1, we obtain a set of colored intervals in the following way: first, identify each binding site $s \in S$ with a color c_s by means of a bijective mapping $c\colon S \to [1 : m]$. Now, introduce an interval $[p + |s| - L : p]$ with color c_s whenever some $s \in S$ occurs at position p in T. We will also refer to the set of colored intervals

$$\{([p + |s| - L : p], c_s) \mid s \text{ occurs at position } p \text{ in } T\}$$

as *the set of intervals induced by S in T with cluster length L*. These intervals are in fact closely related to L-occurrences in T:

Lemma 1 *Let I denote the set of intervals induced by $S = \{s_1, \ldots, s_m\}$ in T with cluster length L. Furthermore, let $A \subseteq S$. Then, the following statements are equivalent:*

(1) There is an integer x such that for all $s \in A$, x stabs an interval in I with color c_s.

(2) A is an L-occurrence in T w.r.t. S.

Proof.
(1)⇒(2): Since x stabs one interval of each color contained in A, x is contained in at least one interval with color c_s, for each $s \in A$. Let $[h_s : i_s]$ denote the corresponding interval with color c_s stabbed by x for $s \in A$. Note that by construction of I, we have $h_s = i_s + |s| - L$ for $s \in A$. Since, by construction of I, s_a occurs at position i_s for each $s \in A$, condition (O1) of an L-occurrence is satisfied, and it remains to prove that condition (O2) holds.

Note that due to $x \in [h_s : i_s]$ for all $s \in A$, we particularly have, for all $s \in A$,

$$x \leq i_s$$

Now, pick $s, t \in A$ arbitrarily. Then $x \in [i_s + |s| - L : i_s]$ implies $x \geq i_s + |s| - L$, and we correspondingly obtain $x \geq i_t + |t| - L$. Putting the latter two inequalities together, we get

$$x \geq \max(i_s + |s|, i_t + |t|) - L. \tag{1}$$

Correspondingly, $x \in [i_t + |t| - L : i_t]$ implies $x \leq i_t$. In an analogous way, we obtain $x \leq i_s$. Putting together the latter two inequalities, we get

$$x \leq \min(i_s, i_t). \tag{2}$$

Add Eqns. (1) and (2), we obtain

$$L \geq \max(i_s + |s|, i_t + |t|) - \min(i_s, i_t).$$

Since we picked s and t arbitrarily, this proves that condition (O2) is satisfied.
(2)⇒(1): Let A be an L-occurrence in T. Then, by condition (O1), for each $s \in A$, there is an index i_s such that s occurs at position i_s in T. Without loss of generality, let

$$x = \min\{i_s \mid s \in A\}. \tag{3}$$

Then, applying (O2), we get

$$|i_s + |s| - x| \leq L.$$

Dropping the absolute value due to $x \leq i_s$, we get $x \geq i_s + |s| - L$. Together with Eq. (3), this yields $x \in [i_s + |s| - L : i_s]$ for all $s \in A$. Since for each s, the latter interval is contained in I with color c_s and is stabbed by x, we are done. □

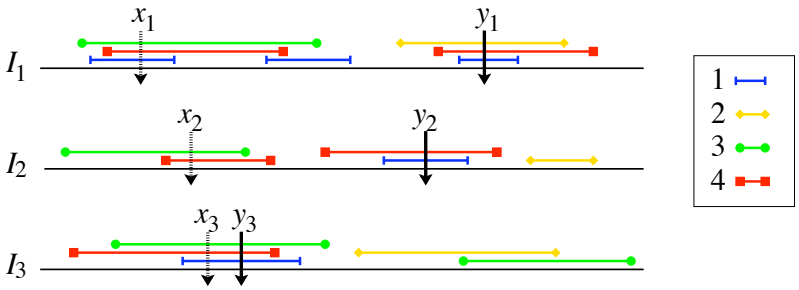

Fig. 2. Example of $(\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3)$-barbecues: The arrows labeled by x_1, x_2, x_3 stab the barbecue $A = \{3, 4\}$, while the arrows labeled by y_1, y_2, y_3 stab the barbecue $B = \{1, 4\}$; both barbecues are best barbecues for this instance.

Given a set of TFBS profiles and a genomic (promoter) sequence, we are particularly interested in L-occurrences of maximum cardinality. Using the above lemma, we can rephrase this problem as maximizing the number of colors that one can stab in an interval arrangement. In fact, this is better illustrated if we assign one of m different barbecue ingredients instead of a color to each interval and identify the string T with a barbecue plate. Then, in order to have a tasty barbecue, our goal is to stab as many different features as possible with a skewer by stabbing only once into the plate. If only one barbecue plate is involved, this constitutes the *single person Best Barbecue Problem*, which can be solved in a straightforward manner.

3 The Best Barbecue Problem

3.1 Interval Barbecues

The Best Barbecue Problem becomes a much more delicate problem if more than one barbecue plate is involved. The idea behind the generalization to K barbecue plates is as follows: suppose we have K guests invited to a barbecue, for each of whom we have prepared one plate with a selection of our m different barbecue ingredients randomly placed on the plate (where the same type of ingredient may be contained an arbitrary number of times on the plate). Now, we want to prepare one skewer for each each guest by stabbing once into each barbecue plate. In order to treat all our guests as equally as possible, the set of ingredients that is contained on all skewers is to be maximized. Note that in addition to the ingredients stabbed on each skewer, some skewers may contain additional features. For an example of the formal definition below, see Fig. 2.

Definition 2 *Let $\mathcal{I}_1, \ldots, \mathcal{I}_K$ denote K sets of intervals, each interval being assigned a color from $[1 : m]$. We say that a set $A \subseteq [1 : m]$ is an $(\mathcal{I}_1, \ldots, \mathcal{I}_K)$-*

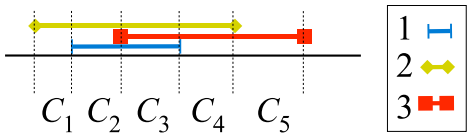

Fig. 3. Obtaining a set system from an interval arrangement, with $C_1 = \{2\}$, $C_2 = \{1, 2\}$, $C_3 = \{1, 2, 3\}$, $C_4 = \{2, 3\}$, and $C_5 = \{3\}$.

barbecue *if for each* $i \in [1 : K]$*, there is an integer* ν_i *such that for each color* $a \in A$*,* ν_i *stabs at least one interval of color* a *in* $\mathcal{I}_i$*.*

A barbecue of maximum cardinality will also be referred to as a best barbecue of $\mathcal{I}_1, \ldots, \mathcal{I}_K$.

This definition immediately suggests to state the following optimization problem, together with the naturally associated decision problem:

Problem 2 Best Barbecue Problem (BBQ)
INSTANCE: Integers m, K*;* $\mathcal{I}_1, \ldots, \mathcal{I}_K$ *denoting* K *sets of intervals, with each interval being assigned a color from* $[1 : m]$*.*
TASK: Determine the maximum cardinality barbecue of $\mathcal{I}_1, \ldots, \mathcal{I}_K$*.*

As for MSLO, we will also be interested in the decision version of the problem, asking whether the best barbecue exceeds a given threshold θ; we will refer to this decision problem as *DBBQ*. Now, the equivalence of arrangements of colored intervals and L-occurrences stated in Lemma 1 tells us that BBQ in fact solves our original problem MSLO.

Beyond our biological problem setting, note that the definition of the Best Barbecue Problem naturally generalizes to colored arrangements of arbitrary geometric objects (such as discs or balls in higher dimension or neighborhoods of vertices in graphs) rather than intervals in one dimension.

3.2 Combinatorial Barbecues

Given a set of colored intervals $\mathcal{I}$, we canonically obtain an equivalence relation between integers – each integer x stabs a certain set of colors in $\mathcal{I}$; we define $x \sim y$ (w.r.t. I) iff x stabs the same set of colors in $\mathcal{I}$ as y does. We refer to the equivalence class of I as the *cells induced by* I (since, in fact, the equivalence classes result from cells of an interval arrangement [31]).

Given K sets of colored intervals $\mathcal{I}_1, \ldots, \mathcal{I}_K$, the cells induced by each $\mathcal{I}_i$ yield a set of subsets of $[1 : m]$. Instead of our original geometric setting, we are

now in a purely combinatorial situation: we only need to work with the sets $\mathcal{C}_1, \ldots, \mathcal{C}_K$, where $\mathcal{C}_i$ denotes the set of cells induced by $\mathcal{I}_i$. As shown in Figure 3, each $C \in \mathcal{C}_i$ is a set of colors, and we have gotten rid of any interval positions. Corresponding to the geometric setting, we say that a set A is a $(\mathcal{C}_1, \ldots, \mathcal{C}_K)$-barbecue iff for each $i \in [1:K]$, there is a $C_i \in \mathcal{C}_i$ such that $A \subseteq C_i$. It is easy to see that every $(\mathcal{I}_1, \ldots, \mathcal{I}_K)$-barbecue is a $(\mathcal{C}_1, \ldots, \mathcal{C}_K)$-barbecue and *vice versa*.

Hence, computing the induced cells for each $\mathcal{I}_i$ leaves us with the following problem:

Problem 3 Combinatorial Best Barbecue Problem (CBBQ)
INSTANCE: Integers m, K*;* $\mathcal{C}_1, \ldots, \mathcal{C}_K$ *denoting* K *sets of subsets of* $[1:m]$*, with* $\lambda_i := |\mathcal{C}_i|$ *and* $\mathcal{C}_i = \{C_{i,1}, \ldots, C_{i,\lambda_i}\}$.
TASK: Maximize

$$|\bigcap_{i \in [1:K]} C_{i,\nu_i}|,$$

with $(\nu_1, \ldots, \nu_K) \in [1:\lambda_1] \times \cdots \times [1:\lambda_K]$.

Corresponding to the decision versions of MSLO and BBQ, we refer to the decision version of CBBQ as DCBBQ. CBBQ has an interesting interpretation in terms of hypergraphs. To establish this connection, we say that a hypergraph with vertices V *supports* a set $Y \subseteq V$ if there is an edge X such that $Y \subseteq X$. Since a each of the K set systems canonically represents a hypergraph, CBBQ simply asks for the largest cardinality edge that is supported in all K hypergraphs.

There are two naive strategies to solve CBBQ (and, correspondingly, DCBBQ):

(A1) Enumerate all $(\nu_1, \ldots, \nu_K) \in [1:\lambda_1] \times \cdots \times [1:\lambda_K]$ and, for each of these vectors, compute $|\bigcap_{i \in [1:K]} C_{i,\nu_i}|$, and keep track of the vector $(\tilde{\nu}_1, \ldots, \tilde{\nu}_K)$ that yields the largest cardinality intersection.

(A2) Enumerate all subsets of $[1:m]$, and, for each subset $A \subseteq [1:m]$, check whether there are suitable indices $\nu_1, \ldots, \nu_K$ such that $A \subseteq \bigcap_{i \in [1:K]} C_{i,\nu_i}$. Keep track of the largest cardinality subset $\tilde{A}$ for which suitable indices were found.

Both of these approaches unfortunately lead to exponential time algorithms – the first algorithm is exponential in K, the second one exponential in m. In fact, we will prove in the next section that DCBBQ is NP-complete. However, since the problem is of practical relevance, we provide branch-and-bound approaches in Section 4.2, implementations of which demonstrate to be useful in some real world instances with limited values for m and K. These will be presented in Section 5. Finally, we provide an algorithm that is exponential in another, rather subtly hidden parameter, which is done in Section 6.

4 Complexity and Algorithms

4.1 NP-completeness Results

Our goal in this section is to prove the following:

Theorem 1

(1) DCBBQ is NP-complete.
(2) DBBQ is NP-complete.
(3) DSLO is NP-complete.

First of all, note that DCBBQ obviously is in NP: given a solution $(\nu_1, \ldots, \nu_K)$, this solution can be trivially verified by computing the cardinality of the intersection $|\bigcap_i C_{i,\nu_i}|$ in $O(mK)$ time. An analogous argument shows that DBBQ and also DSLO is in NP.

Our proof of NP-completeness works by reducing the problem of deciding whether a K-partite graph contains a K-clique to DCBBQ. Let $G = (V, E)$ denote an undirected K-partite graph, i.e., we have $V = V_1 \cup \cdots \cup V_K$ as the disjoint union of the layers V_i and $|V_i \cap e| \leq 1$ for any $i \in [1:K]$ and $e \in E$ (writing edges of G as two-element subsets of V). A *K-clique* in G is a set of vertices $v_1, \ldots, v_k$ with $v_i \in V_i$ and $\{v_i, v_j\} \in E$ for all i, j. As has been noted by several authors and formally proved by Azarenok *et al.*, the following holds:

Lemma 2 *([4]) Deciding whether a K-partite graph has a K-clique is NP-complete.*

Given a K-partite graph G, we now construct a collection $\mathcal{C}_1, \ldots, \mathcal{C}_K$ of subsets of $[1:m]$ such that there is a barbecue of cardinality K iff G has a K-clique. We start with defining the neighborhood set of a vertex v as

$$N(v) := \{\, w \in V \mid \{v, w\} \in E \,\}$$

for $v \in V$. Furthermore, for $v \in V$, define $C_v := N(v) \cup \{v\}$. The following Lemma establishes close connections between the graph G and intersections of the sets C_v (i.e., edges shared by the K hypergraphs):

Lemma 3 *For a K-partite graph $G = (V, E)$, let $v_1 \in V_1, \ldots, v_K \in V_K$, where V is the disjoint union of $V_1, \ldots, V_K$. The following holds:*

(1) $\{u, v\} \in E \iff \{u, v\} \subseteq C_u \bigcap C_v$,
(2) $\bigcap_{i \in [1:K]} C_{v_i} \subseteq \{v_1, \ldots, v_K\}$,
(3) $|\bigcap_{i \in [1:K]} C_{v_i}| = K \iff$ G has a K-clique.

Proof. (1): Let $\{u, v\} \in E$. Then, by construction, we have $u \in C_u$ and $u \in N(v)$, and hence also $u \in C_v$. Analogously, $v \in C_v$ and $v \in N(u)$ yields $v \in C_u$, so that we have $\{u, v\} \subseteq C_u \cap C_v$.
Conversely, let $\{u, v\} \subseteq C_u \cap C_v$. Then, $v \in C_u$ implies $v \in N(u)$, and hence $\{u, v\} \in E$.
(2): Let $x \in \bigcap_{i\in[1:K]} C_{v_i}$, and assume that $x \notin \{v_1, \ldots, v_K\}$. Furthermore, w.l.o.g, assume that $x \in V_1$. Then, in particular, we have $x \in C_{v_1}$. Now, by construction, the only vertex from V_1 contained in C_{v_1} is v_1 itself. However, we assumed that $v_1 \neq x \in C_{v_1}$, which is a contradiction.
(3): Let $|\bigcap_{i\in[1:K]} C_{v_i}| = K$. Then claim (2) implies that $\bigcap_{i\in[1:K]} C_{v_i} = \{v_1, \ldots, v_K\}$. It remains to be shown that $\{v_i, v_j\} \in E$ for all $i, j \in [1 : K]$. To this end, observe that we have $\{v_i, v_j\} \in C_{v_1} \cap C_{v_2}$. Using part (1) of this Lemma, this implies $\{v_i, v_j\} \in E$.
Conversely, let $\{v_1, \ldots, v_K\}$ be a K-clique in G. Then, for arbitrary $i, j \in [1 : K]$, we have $v_i \in N(v_j)$, and hence $v_i \in C_{v_j}$. By construction, we also have $v_i \in C_{v_i}$. Altogether, we obtain $\{v_1, \ldots, v_K\} \subseteq \bigcap_{i\in[1:K]} C_{v_i}$, implying $|\bigcap_{i\in[1:K]} C_{v_i}| \geq K$. Claim (2) immediately implies $|\bigcap_{i\in[1:K]} C_{v_i}| \leq K$, so that we have $|\bigcap_{i\in[1:K]} C_{v_i}| = K$. □

Proof of Theorem 1. We start with the *proof of (1).*

Since choosing $\mathcal{C}_i := \{C_{v_i} \mid v_i \in V_i\}$ for all $i \in [1 : K]$ together with $\theta := K$ gives us an instance of the combinatorial barbecue decision problem, part (3) of Lemma 3 reduces the decision problem whether a K-partite graph has a K-clique to the combinatorial barbecue decision problem. Since the construction can be performed in polynomial time, this immediately yields the desired NP-completeness proof.

Proof of (2): Our proof works by reducing DCBBQ to DBBQ. Let $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ denote the sets of subsets corresponding to an instance of DCBBQ. Given $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$, we construct a set of interval sets $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ that constitute an instance of DBBQ satisfying

$$A \text{ is a } \mathcal{C}\text{-barbecue} \iff A \text{ is an } \mathcal{I}\text{-barbecue} \tag{4}$$

for any $A \subseteq [1 : m]$. To this end, let $\mathcal{C}_i = \{C_{i,1}, \ldots, C_{i,\lambda_i}\}$. For each $\mu \in [1 : \lambda_i]$, we a set of colored intervals $\mathcal{I}_{i,\mu}$ as follows:

$$\mathcal{I}_{i,\mu} := \{([\mu, \mu], j) \mid j \in C_{i,\mu}\}. \tag{5}$$

Now, choosing

$$\mathcal{I}_i := \bigcup_{1\leq\mu\leq\lambda_i} \mathcal{I}_{i,\mu} \tag{6}$$

for each $i \in [1:K]$ yields an instance of DBBQ. It remains to show that this instance satisfies the equivalence from Eq. (4). Let A be a $\mathcal{C}$-barbecue. By Definition 2, there are indices $\nu_1, \ldots, \nu_K$ such that $A \subseteq C_{i,\nu_i}$. Looking at Eq. (5), ν_i stabs each $j \in C_{i,\nu_i}$ in $\mathcal{I}_{i,\nu_i}$, so that in particular, ν_i stabs each $j \in A$ in $\mathcal{I}_{i,\nu_i}$. Now, Eq. (6) implies $\mathcal{I}_{i,\nu_i} \subseteq \mathcal{I}_i$, so that ν_i stabs each $j \in A$ in $\mathcal{I}_i$. Hence, A is an $\mathcal{I}$-barbecue.

Conversely, let A be an $\mathcal{I}$-barbecue, and let $\nu_1, \ldots, \nu_K$ denote the corresponding indices such that ν_i stabs A in $\mathcal{I}_i$. Note that the intervals stabbed by ν_i in $\mathcal{I}_i$ are precisely those that are contained in $\mathcal{I}_{i,\nu_i}$. By construction, $\mathcal{I}_{i,\nu_i}$ contains one an interval of each color contained in C_{i,ν_i}, so that each color that is stabbed by ν_i in $\mathcal{I}_i$ is contained in C_{i,ν_i}, in other words, we have $A \subseteq C_{i,\nu_i}$ for each $1 \leq i \leq K$, so that A is a (combinatorial) $\mathcal{C}$-barbecue.

Eq. (4) obviously reduces CBBQ to BBQ. Furthermore our construction can clearly be performed in polynomial time. Thus the proof of claim (2) is complete.

Proof of (3): Analogous to the proof of claim (2), we reduce DCBBQ to DSLO and start with constructing a string $T_{i,\mu} \in \Sigma^*$ from each $C_{i,\mu}$, with $\Sigma := \{\alpha_0, \ldots, \alpha_m\}$. To this end, let $C_{i,\mu} = \{j_1, \ldots, j_p\}$, so that we can write

$$T_{i,\mu} := \alpha_0^{2m-p} \alpha_{j_1} \ldots \alpha_{j_p}.$$

This allows us to define T_i as the concatenation of all $T_{i,\mu}$, i.e.,

$$T_i := T_{i,1} \ldots T_{i,\lambda_i}.$$

Now, choosing $L := m + 1$ and $S := \{\alpha_1, \ldots, \alpha_m\}$, it remains to be shown that for any $A = \{j_1, \ldots, j_p\} \subseteq [1:m]$ and $A' = \{\alpha_{j_1}, \ldots, \alpha_{j_p}\} \subseteq [1:m]$

$$A \text{ is a } \mathcal{C}\text{-barbecue} \iff A' \text{ is an } L\text{-occurrence of } S \text{ in } T_1, \ldots, T_K. \tag{7}$$

To see this, let A be a $\mathcal{C}$-barbecue with corresponding indices $\nu_1, \ldots, \nu_K$. If we write $C_{i,\nu_i} = \{j_1, \ldots, j_p\}$, then by construction each T_{i,ν_i} contains the string $\alpha_{j_1} \ldots \alpha_{j_p}$, which constitutes an $(m+1)$-occurrence of $\{\alpha_{j_1}, \ldots, \alpha_{j_p}\}$. Since A is a subset of C_{i,μ_i}, in particular T_{i,ν_i} contains an $(m+1)$-occurrence of A'. Finally, T_{i,ν_i} is a substring of T_i, so that in particular T_i contains an $(m+1)$-occurrence of A'.

Conversely, let A' be an $(m+1)$-occurrence of A' in $T_1, \ldots, T_K$. Since each of the blocks $T_{i,\mu}$ starts with α_0^m, the $(m+1)$-occurrence of A' in T_i is contained within one single block, there is a unique index ν_i such that A' is an $(m+1)$-occurrence in T_{i,ν_i}. The corresponding set C_{i,ν_i} that T_{i,ν_i} was constructed from hence contains A as a subset, such that A is a $\mathcal{C}$-barbecue. $\square$

Studying the algorithm specified in the last paragraph of Section 3.2 in more detail, one realizes that the branch-and-bound principle can be applied in the following way: Suppose we have already found a vector $(\tilde{\nu}_1, \ldots, \tilde{\nu}_K)$ such that $|\bigcap_{i\in[1:K]} C_{i,\tilde{\nu}_i}| = \theta$. Now, when enumerating index vectors $(\nu_1, \ldots, \nu_K)$, we start with picking ν_1, then we pick ν_2, and so on. If at some point, we have picked $\nu_1, \ldots, \nu_a$ (with $a < K$), and we find that $\bigcap_{i\in[1:a]} C_{i,\nu_i} \leq \theta$, we know that no matter how we choose $\nu_{a+1}, \ldots, \nu_K$, the cardinality of the intersection $\bigcap_{i\in[1:K]} C_{i,\nu_i}$ cannot exceed θ. In terms of a branch-and-bound algorithm, this means that if t denotes the cardinality of the best barbecue so far, then $|\bigcap_{i\in[1:a]} C_{i,\nu_i}| \leq t$ is an upper-bound-criterion for the set of all instances $\{(\nu_1, \ldots, \nu_a, \mu_{a+1}, \ldots, \mu_K) \mid \mu_i \in [1 : \lambda_1]\}$. Whenever the upper bound is smaller than the best solution so far, this set of instances can be ignored by the algorithm.

Concerning time complexity, note that computing the intersection of K subsets of $[1 : m]$ can be done in $O(Km)$ time. Hence, it can be seen easily that Algorithm (A1) (as well as the branch-and-bound version) takes $O(Km\lambda^K)$ time, where λ denotes the maximum of all of all λ_i. In practice, the branch-and-bound version of Algorithm (A1) applied to the phylogenetic footprinting problem can be observed to yield a significant speed-up.

We now turn to algorithm (A2), which can also be improved using a branch-and-bound-like approach. Observe that if $A \subseteq [1 : m]$ is not an $(\mathcal{I}_1, \ldots, \mathcal{I}_K)$-barbecue, then all sets A' with $A \subseteq A'$ are not barbecues either. In particular, sets that are not barbecues cannot be best barbecues. In terms of a branch-and-bound algorithm, this means that if we encounter a set A that is not a barbecue, we do not need to examine the set of instances

$$\{A' \subseteq [1 : m] \mid A \subseteq A'\}.$$

As another improvement for Algorithm (A2), note that not necessarily all subsets of $[1 : m]$ need to be enumerated – one can limit the algorithm to consider only sets $A \subseteq [1 : m]$ such that some set A' with $A' \supseteq A$ is contained in at least one $\mathcal{C}_i$. Finally, it is easy to see that, with $\Lambda := |\mathcal{C}_1| + \cdots + |\mathcal{C}_K|$, the running time of Algorithm (A2) is $O(2^m \Lambda m)$.

5 Computational Example

As an illustrative example for the application of the BBQ approach to biological data we consider here a short region selected from the *Hox* clusters.

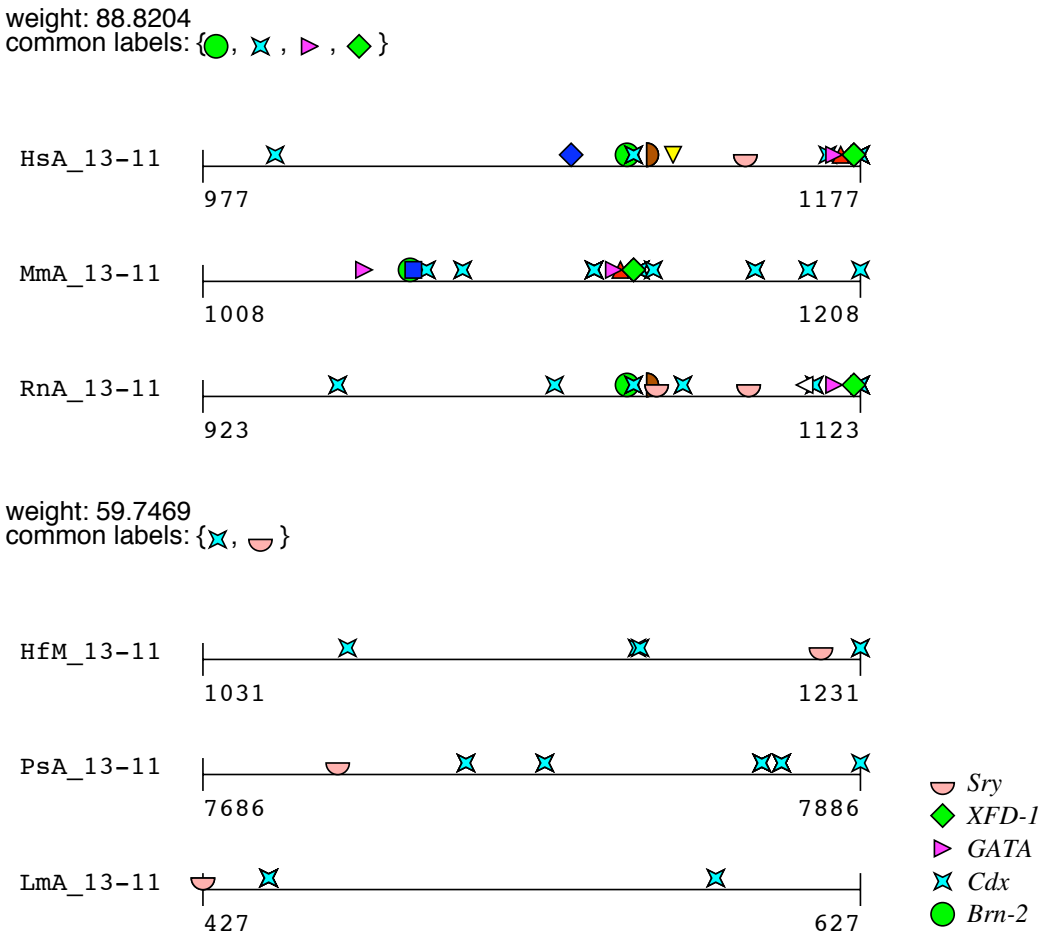

Fig. 4. A significant cluster of binding sites among evolutionary closely related species (placental mammals: Hs Human, Mm Mouse, and Rn Rat) (upper panel). Searching for the same set of candidate binding sites in evolutionary more distant vertebrates (Hf shark, Ps bichir, Lm coelacanth), one obtains a smaller and hence less significant — probably non-functional — cluster (lower panel). Both clusters were obtained with the weighted version of Algorithm (A2).

The *Hox* genes form a class of homeodomain transcription factors and have a crucial role in early embryonic development [20]. In vertebrates, these genes are located within tightly linked gene clusters. We focus here on the intergenic region between *HoxA13* and *HoxA11*, which has a length between 12000 and 15000 nucleotides. The particular locus includes the promoter region of *HoxA11* and is important for the development of the limb bud, see e.g. [38].

In order to select binding motifs, we can either use databases of known transcription factor binding sites such as TRANSFAC [15] or JASPAR [28] or derive the motifs from phylogenetic footprinting [27,9] or statistical local alignment procedures [16]. For our example, we used a comparatively small selection of $m = 15$ binding site profiles predicted to match a conserved non-coding

region in tetrapods using `tfsearch` and `TRANSFAC`, release 3.3.

A `C++`-implementation of Algorithm (A2) took only a few seconds of computation time on a standard desktop computer with a 2.8 MHz processor [1] to produce the results in Fig. 4 for $K = 3$, $m = 15$, $L = 200$, and genomic input sequences with a length between 12000 and 15000.

The best CRM within the *HoxA13-HoxA11* region hits the selected conserved non-coding region and contains five common labels. Three of these are exclusively shared among the selected tetrapods, here only placental mammals (upper panel in Fig. 4).

The "fishes", which branch off before the origin of tetrapods, neither share the conserved non-coding region nor this CRM, i.e., these five binding sites do not appear together in these species, not even in a different order. Conceivably, this CRM could be associated with one of the major innovations involving the adjacent genes, e.g. the fin-limb transition or emergence of the placenta.

Using the bounded difference method from Section 6 with $\delta = 2$ as bound, even instances with $K = 5$ and $m = 300$ can be computed in less than one minute. Note that the implementation supports several features that are useful in practise. For instance, highly correlated binding site profiles can be treated as a group of binding sites (so that overlapping occurrences are counted as a single occurrence). The current implementation also supports the weighting schemes proposed in Section 6.3.

6 Variants of the Best Barbecue Problem

The mostly theoretical results presented in the previous sections should rather be seen as a foundation for practically relevant variations and extensions. The implementations and results indicate that the approach is principally suited for practical applications; yet the basic problem setting needs to be adapted so that the discovery procedure takes into account common effects such as binding-site turnover discussed above.

6.1 Barbecues with Limited Support

In the problem setting as discussed so far, regulatory modules are expected to occur in *all* sequences involved. While this is useful when comparing few

[1] The source code of our implementation is available for download at `http://www.bioinf.uni-leipzig.de/Software/bbq/`

evolutionarily related promoter sequences, this is eventually not a reasonable model when dealing with promoters of many co-expressed genes from one species. In this scenario, regulatory modules are rather expected to occur in few of the promoter sequences under consideration. Also, binding-site turnover may impose limitations even in the case of evolutionarily related sequences.

Consequently, it makes sense to introduce an extra *support parameter* σ to the best barbecue problem (which carries canonically to MSLO): we seek a binding site set B of maximum cardinality that occurs in at least σ of the K set systems. Since this is a more general problem than the best barbecue problem, the NP-completeness results still hold for this problem; branch-and-bound algorithms for this extended problem can eventually be derived from the ones discussed in Section 4.2. The support parameter provides a very interesting link to *frequent itemset mining* [2], which certainly is one of the most important concepts from data mining: if all K set systems consist of one set only, then the best barbecue with support σ is a *maximum frequent itemset with support* σ. Hence, the best barbecue problem with limited support is a – quite natural – generalization of frequent itemset mining: while in frequent itemset mining, we are given K sets of items, the best barbecue problem deals with K *sets of sets* of items. If the sets in each of the K set systems represent co-localized items, we obtain a notion of "frequent co-localized itemsets", where co-localization can be derived from a suitable "topology" of the space in which the items occur – in the case of regulatory modules, this topology is given by distances between occurrences of binding sites.

In a conceptually similar approach, one may introduce a scoring function that measures the similarity between CRMs in terms of their constituent TFBS. A natural choice for this purpose is the so-called Tanimoto score for measuring the (dis)similarity between sets of objects, in our case occurrences of binding sites. One then searches the input sequences for collections of TFBS that are sufficiently similar. A major advantage of using Tanimoto scores is that they not require the specification of an additional support parameter. This approach is explored in detail in [21].

6.2 Bounded Differences

In the barbecue-party illustration of our optimization problem, the optimal solution may sometimes appear rather unfair: although all guests share a maximum number of equal ingredients, some guests might get a large number of extra ingredients, while others get no extra ingredients at all. To treat our guests more equally, we might consider to limit the number of extra features. This limitation, in fact, has further advantages: first of all, the computational complexity of the problem is reduced – the algorithms we obtain will turn

out to be exponential in the maximum number of extra features rather than the overall number of features. Secondly, bounding the number of extra features makes sense in our biological problem setting: if there is a large number of extra features within a *cis*-regulatory module, this means that within one footprint cluster, a large number of "foreign" binding sites is present, so that the function of the relevant binding sites might well be disturbed.

In our formal problem setting, we restrict ourselves to considering combinatorial best barbecues with a bounded number of extra features; our considerations, however, carry naturally to geometric barbecues as well as to L-occurrences.

Suppose we are given an instance of the combinatorial best barbecue problem, together with a barbecue $B = \bigcap_i C_{i,\nu_i}$. The number of "extra" features occurring in C_{i,ν_i} now reads as $|C_{i,\nu_i} \setminus B|$. Correspondingly, we say that B is a *δ-bounded barbecue* if there are indices $\nu_1, \ldots, \nu_K$ such that, for each i, $|C_{i,\nu_i} \setminus B| \leq \delta$. We now consider δ as an additional input parameter and want to compute the largest cardinality δ-bounded barbecue. Observe that for $\delta = 0$ and given an arbitrary $B \subseteq [1:k]$, we can check in $O(Km \log \Lambda)$ time whether B is a 0-bounded barbecue. To see this, note that we merely need to check whether $B \in \mathcal{C}_i$ for each i. Clearly, this can be done using binary search by canonically identifying a subset X of $[1:m]$ with a number between 0 and $2^m - 1$ (where the j-th bit is 1 iff $j \in X$). Since each comparison during our binary search takes $O(m)$ time, we obtain the running time claimed above.

Now, computing the largest cardinality 0-bounded barbecue is easy: we test for each $B \in \mathcal{C}_1$ and for each $i \in [1:K]$ whether $B \in \mathcal{C}_i$. If for some B, we have $B \in \mathcal{C}_i$ for all $i \in [1:K]$, we check whether $|B|$ exceeds the largest solution found so far. Doing so for all $B \in \mathcal{C}$ yields the largest cardinality 0-bounded barbecue. Since we test $|\mathcal{C}| = \Lambda$ many sets B whether B is a 0-bounded barbecue, so that the overall running time amounts to $O(\Lambda K m \log \Lambda)$.

This idea carries to finding largest cardinality δ-bounded barbecues. Each of the sets $\mathcal{C}_i$ needs to be supplemented as follows:

$$\mathcal{C}'_i := \bigcup_{A \in \mathcal{C}_i} \bigcup_{D \in P_\delta(A)} A \setminus D.$$

Here, $P_\delta(A)$ denotes the set of all subsets of A whose cardinality is at most δ. The algorithm for finding largest cardinality δ-bounded barbecues now works the same way as the algorithm for 0-bounded barbecues, with $\mathcal{C}_1, \ldots, \mathcal{C}_K$ substituted by $\mathcal{C}'_1, \ldots, \mathcal{C}'_K$ and $\mathcal{C}$ substituted by $\mathcal{C}' := \mathcal{C}'_1, \ldots, \mathcal{C}'_K$. Since each $|\mathcal{C}'_i|$ is bounded by $m^\delta |\mathcal{C}_i|$, we obtain a running time of $O(m^\delta \Lambda K m \delta \log(m\Lambda))$.

6.3 Weighted Versions

As a final useful and practically relevant extension, we provide a basis to find maximum *weighted* barbecues. Given a (finite) set M, we define a *weighted subset of* M as a mapping $A\colon M \to \mathbb{R}_{\geq 0}$. Now, given $A, B\colon M \to \mathbb{R}_{\geq 0}$, we define $A \cap B\colon M \to \mathbb{R}_{\geq 0}$ by

$$(A \cap B)(i) := \begin{cases} A(i) + B(i) & \text{if } A(i)B(i) \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

The algorithms discussed naturally generalize to the weighted version of the best barbecue problem resulting canonically from weighted subsets and their intersections. In fact, the practical results discussed in the following section were obtained with an implementation of such a weighted version. The weighted version is particularly useful in practice if the candidate binding sites are given in the form of so-called position weight matrices, which are available in typical binding site databases [15,28]. Using position weight matrices, each occurrence of binding site s_j is associated with a weight between 0 and 1.

7 Concluding Remarks

Summarizing our results, we have shown that a natural approach to the discovery of *cis*-regulatory modules leads to an elementary optimization problem, which we have shown to be computationally hard in general. Also, a slight and still natural generalization of this setting leads to a problem that also is a generalization of the well-established concept of frequent itemset mining.

We provide an illustrative example of regulatory modules in *Hox* gene promoters, obtained using an implementation of the branch-and-bound algorithms we propose. The results pose a good perspective for a systematic study comparing our results with the outcome of related or alternative approaches. Such an endeavour, however, requires the careful preparation of both real and artificial benchmark data sets and the design of clear rules how different programs with different requirements on their input data can be fairly compared. Such a benchmark study thus goes beyond the purpose of this contribution.

Acknowledgments. We thank Andreas Dress for helpful comments and Manja Marz for computational assistance. This work was supported in part by the *DFG* Bioinformatics Initiative BIZ-6/1-2.

References

[1] S. Aerts, P. Van Loo, Y. Moreau, and B. De Moor. A genetic algorithm for the detection of new cis-regulatory modules in sets of coregulated genes. *Bioinformatics*, 20:1974–1976, 2004.

[2] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, 22:207–216, 1993.

[3] M. I. Arnone and E. H. Davidson. The hardwiring of development: Organization and function of genomic regulatory systems. *Development*, 124:1851–1864, 1997.

[4] A. Azarenok and V. Krikun. A clique in a n-partite graph and optimal orientation of functional blocks of integral schemes *(in Russian). Izv. Akad. Nauk BSSR, Ser. Fiz.-Mat. Nauk (Proc. Ac. Sc. Belarus. Phys.-Math. Ser.)*, 2:8–15, 1988. Zentralblatt MATH Accession Number Zbl 0652.05057.

[5] M. P. Beal, A. Bergeron, S. Corteel, and M. Raffinot. An algorithmic view of gene teams. *Theoretical Computer Science*, 320:395–418, 2004.

[6] S. Ben-Tabou de Leon and E. D. Davidson. Gene regulation: Gene control network in development. *Ann. Rev. Biophys. Biomol. Struct.*, 36:191–212, 2007.

[7] B. P. Berman, B. D. Pfeiffer, T. R. Laverty, S. L. Salzberg, G. M. Rubin, M. B. Eisen, and S. E. Celniker. Computational identification of developmental enhancers: conservation and function of transcription factor binding-site clusters in *Drosophila melanogaster* and *Drosophila pseudoobscura*. *Genome Biol.*, 5:R61, 2004.

[8] M. Blanchette, A. R. Bataille, X. Chen, C. Poitras, J. Laganiere, C. Lefebvre, G. Deblois, V. Giguere, V. Ferretti, D. Bergeron, et al. Genome-wide computational prediction of transcriptional regulatory modules reveals new insights into human gene expression. *Genome Research*, 16:656, 2006.

[9] M. Blanchette and M. Tompa. `FootPrinter`: a program designed for phylogenetic footprinting. *Nucleic Acids Res.*, 31:3840–3842, 2003.

[10] N. A. Chuzhanova, M. Krawczak, L. A. Nemytikova, V. D. Gusev, and D. N. Cooper. Promoter shuffling has occurred during the evolution of the vertebrate growth hormone gene. *Gene*, 254:9–18, 2000.

[11] T. E. P. Consortium. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, 447:799–816, 2007.

[12] E. Davidson. *Genomic Regulatory Systems*. Academic Press, San Diego, 2001.

[13] S. W. Doniger and J. C. Fay. Frequent gain and loss of functional transcription factor binding sites. *PLoS Comp. Biol.*, 3:e99, 2007.

[14] A. Erives and M. Levine. Coordinate enhancers share common organizational features in the drosophila genome. *Proc. Natl. Acad. Sci. USA*, 101:3851–3856, 2004.

[15] T. Heinemeyer, E. Wingender, I. Reuter, H. Hermjakob, A. E. Kel, O. V. Kel, E. V. Ignatieva, E. A. Ananko, O. A. Podkolodnaya, F. A. Kolpakov, N. L. Podkolodny, and N. A. Kolchanov. Databases on transcriptional regulation: TRANSFAC, TRRD, and COMPEL. *Nucleic Acids Res.*, 26:364–370, 1998.

[16] J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol*, 296:1205–1214, 2000.

[17] O. V. Kel-Margoulis, T. G. Ivanova, E. Wingender, and A. E. Kel. Automatic annotation of genomic regulatory sequences by searching for composite clusters. In *Proc. Pac. Symp. Biocomput.*, pages 187–198, 2002.

[18] J. Kim, J. Seo, Y. S. Lee, and S. Kim. `TFExplorer`: integrated analysis database for predicted transcription regulatory elements. *Bioinformatics*, 21:548–550, 2005.

[19] M. Z. Ludwig, C. Bergman, N. H. Patel, and M. Kreitman. Evidence for stabilizing selection in a eukaryotic enhancer element. *Nature*, 403:564–567, 2000.

[20] W. McGinnis and R. Krumlauf. Homeobox genes and axial patterning. *Cell*, 68:283–302, 1992.

[21] P. Menzel, P. F. Stadler, and A. Mosig. Tanimoto's Best Barbecue: Discovering regulatory modules using tanimoto scores. In *GCB 2007*, Lecture Notes in Informatics, 2007. in press.

[22] A. M. Moses, D. A. Pollard, D. A. Nix, V. N. Iyer, X. Y. Li, M. D. Biggin, and M. B. Eisen. Large-scale turnover of functional transcription factor binding sites in *Drosophila*. *PLoS Comp. Biol.*, 2:e130, 2006.

[23] K. Noto and M. Craven. Learning probabilistic models of cis-regulatory modules that represent logical and spatial aspects. *Bioinformatics*, 23:e156–e162, 2007.

[24] E. Pennisi. Searching for the genome's second code. *Science*, 306:632–635, 2004.

[25] P. Perco, A. Kainz, G. Mayer, A. Lukas, R. Oberbauer, and B. Mayer. Detection of coregulation in differential gene expression profiles. *Biosystems*, 82:235–247, 2005.

[26] A. Philippakis, F. He, and M. Bulyk. Modulefinder: a tool for computational discovery of cis regulatory modules. In *Proc. Pac. Symp. Biocomput.*, pages 519–30, 2005.

[27] S. Prohaska, C. Fried, C. Flamm, G. Wagner, and P. Stadler. Surveying phylogenetic footprints in large gene clusters: Applications to Hox cluster duplications. *Mol. Phyl. Evol.*, 31:581–604, 2004.

[28] A. Sandelin, W. A. Pär Engström, W. Wasserman, and B. Lenhard. JASPAR: an open access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.*, 32:D91–D94, 2004.

[29] R. Sanges, E. Kalmar, P. Claudiani, M. D'Amato, F. Muller, and E. Stupka. Shuffling of cis-regulatory elements is a pervasive feature of the vertebrate lineage. *Genome Biol.*, 7:R56, 2006.

[30] R. Sharan, A. Ben-Hur, G. G. Loots, and I. Ovcharenko. `CREME`: Cis-regulatory module explorer for the human genome. *Nucleic Acids Res.*, 32:W253–W256, 2004.

[31] M. Sharir and P. Agarwal. Arrangements and their applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 49–119. North-Holland, New York, 2000.

[32] S. Sinha, E. van Nimwegen, and E. D. Siggia. A probabilistic method to detect regulatory modules. *Bioinformatics*, 19:i292–i301, 2003.

[33] A. D. Smith, P. Sumazin, and M. Q. Zhang. Tissue-specific regulatory elements in mammalian promoters. *Mol. Systems Biol.*, 3:73, 2007.

[34] A. Tanay, A. Regev, and R. Shamir. Conservation and evolvability in regulatory networks: the evolution of ribosomal regulation in yeast. *Proc. Natl. Acad. Sci. USA*, 102:7203–7208, 2005.

[35] W. W. Wassermann and A. Sandelin. Applied bioinformatics for the identification of regulatory elements. *Nat. Rev. Genet.*, 5:276–287, 2004.

[36] G. A. Wray, M. W. Hahn, E. Abouheif, J. P. Balhoff, M. Pizer, M. V. Rockman, and L. A. Romano. The evolution of transcriptional regulation in eukaryotes. *Mol. Biol. Evol.*, 20:1377–1419, 2003.

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

[38] J. Zakany and D. Duboule. The role of *Hox* genes during vertebrate limb development. *Curr. Opin. Genet. Dev.*, 2007. doi:10.1016/j.gde.2007.05.011.