Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Detecting Phylogenetic Footprint Clusters by Optimizing Barbeques

by

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

Preprint no.: 32 2005

Detecting Phylogenetic Footprint Clusters by Optimizing Barbeques

Axel Mosig[1], Türker Bıyıkoğlu[2], Sonja J. Prohaska[1], Peter F. Stadler[1]

[1]Bioinformatik, Institut für Informatik, Universität Leipzig, Kreuzstraße 7b, D-04103 Leipzig, Germany Phone: ++49 341 149 5120; Fax: ++49 341 149 5119;
Email: axel@bioinf.uni-leipzig.de.
[2]Max Planck Institute for Mathematics in the Sciences, Inselstrasse 22, D-04103 Leipzig, Germany

Abstract. Taking a geometric view on a problem occurring in the context of phylogenetic footprinting, we study the so-called *Best Barbeque Problem*. The Best Barbeque Problem asks for simultaneously stabbing a maximum number of differently colored intervals from K arrangements of colored intervals. This geometric problem leads to a combinatorial optimization problem, a decision version of which is shown to be NP-complete. Due to its relevance in biological applications, we propose algorithmic variations that are suitable for detecting footprint clusters in some real world instances of phylogenetic footprints.

1. Introduction

Alignment algorithms take an important role in many parts of bioinformatics research and its applications. Aligning nucleotide sequences with established state-of-the-art algorithms (see [10]) is a major ingredient of studying evolutionary development, genome annotation, binding site identification or phylogenetic footpriting, and numerous other applications in biology. Most alignment algorithms deal with aligning sequences and are usually founded on dynamic programming techniques. Such algorithms typically construct order-preserving mappings between sequence segments. If a variable number rather than two sequences are to be aligned, the underlying longest common subsequence is known to be NP-complete [8].

The biological motivation of alignment tasks usually is to discover what elements have been conserved due to a certain function exhibited by these elements. The problem that we deal pursuits the same purpose, namely discovering elements that have been conserved for functional reasons. In our setting, the underlying structures (and hence the resulting algorithms), however, differ substantially from the usual notion of alignments: since the elements that we deal with, i.e., transcription factor binding sites (occuring as short fragments on genomic sequences), exhibit their function independent of their order of occurence along a genomic sequence, but rather by occuring clustered within a genomic region of limited size, we are confronted with a non-sequential alignment problem. This leads to novel algorithms that we present in this work. As it turns out, the problem we deal with in general is NP-complete in a similar way as the longest common subsequence problem becomes NP-complete if a variable number of sequences is involved. However, we identify algorithms whose time and memory demand is exponetial in (rather subtly hidden) parameters that are small in practice, and hence we obtain algorithms that are practical for most instances that occur in biological applications.

The outline of this paper is as follows: In the following two sections, we give a formal problem description, and provide the basic ideas of the biological relevance of the problem we deal with. 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 Barbeque Problem* which deals with simultaneously stabbing intervals of the same color from several interval arrangements and can be rephrased as a combinatorial optimization problem. In Section 5.1, we show that this combinatorial version of the Best Barbeque Problem is NP-complete. We then provide branch-and-bound-like algorithms, with some results from a biological application demonstrating the practical relevance of the problem. Each of these algorithms is exponential in a different input parameter, hence the algorithms are useful for different types of instances.

The problem of detecting *cis*-regulatory modules has been adressed only very recently, see [11, 6]. As opposed to the approaches investigated in previous research, our approach carries the concept behind multiple alignments into discovering *cis*-regulatory modules, founded on a novel type of algorithms.

2. Biological Background

Understanding the mechanisms of gene expression is a major challenge of current genomics. Transcription in eukaryotic cells is regulated by a complex assembly of proteins that specifically bind to the DNA. Indeed, experimental evidence from a variety of sources shows that a major mode of developmental gene evolution is based on the modification of "cis-regulatory elements", i.e., DNA motifs that are recognized by components of the transcription complex [3]. The investigation of the molecular evolution of these cis-regulatory elements is difficult because of the absence of a reliable "genetic code for non-coding sequences". Binding sites for transcription factors are usually short and variable and are thus hard to identify unambiguously, in particular if the transcription factors involved are not known *a priori* [14, 7]. It has been noted for a long time, however, that non-coding sequences can contain islands of strongly conserved segments, so-called *phylogenetic footprints* [13]. In many cases phylogenetic footprints have experimentally been shown to be indicative of functional cis-regulatory elements, see e.g. the reviews [4, 5].

Phylogenetic footprints are almost always detected in clusters that comprise multiple transcription factor binding sites, each of which is often less than 10 nucleotides long. In order to be functional, neither the order nor the the orientation of these individual binding sites is relevant, but merely the fact that they occur clustered. While order and orientation are typically conserved in homologous genes (i.e., for the same gene in different species), this is not necessarily true for genes within the same organism that are nevertheless regulated by the same combination of transcription factors. The problem that biologists need to solve in this context therefore is to find a maximum set of short sequence fragments that occur clustered (i.e., close to each other) on several large genomic sequence fragments.

3. L-occurences and Interval Arrangements

Throughout this paper, let Σ denote some finite alphabet. When dealing with genome sequences, we usually have $\Sigma = \{C, G, T, A\}$ 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 between a and 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 $S = \sigma_1 \ldots \sigma_n$, let $[S]$ denote its length and for any two integers a, b we write $S|_{a,b}$ for the substring $\sigma_a \sigma_{a+1} \ldots \sigma_b$. 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, footprint regions, i.e., evolutionary conserved sequence parts, are *clustered* occurences of short fragments along a genome. We formally grasp the notion of clustered occurences by introducing a cluster length L and say that fragment occurrences are (L-)clustered if the occurences are contained within an interval of size L along the genome:

Definition 1. *Let* $S = \{s_1, \ldots, s_m\}, T \in \Sigma^*$, $L \in \mathbb{N}$ *and* $A \subseteq S$ *with* $|A| =: \ell$. *We say that* A is an L-occurence in T w.r.t. S *if there is a mapping* $i\colon A \to \mathbb{N}$ *associating an index* i_s *with each* $s \in A$ *such that*

(O1) s *occurs in* T *at position* i_s *for each* $s \in A$ *and*
(O2) $|i_s + [s] - i_t| \leq L$ *for all* $s, t \in A$.

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

In general, we are interested in finding L-occurences of maximum cardinality.

However, before we turn to the problem that is relevant for phylogenetic footprint cluster detection – namely finding L-occurences that can be found simultaneously in several genomes $T_1, \ldots, T_K$ treated in Section 4 – we study the scenario involving a single sequence T. 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 fragment $s \in S$ 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-occurences in T:

Lemma 2. *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-occurence 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-occurence 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. We distinguish two cases, starting with $i_s \leq i_t$. Then $x \in [i_s + [s] - L : i_t]$ implies

$$x \geq i_s + [s] - L. \tag{1}$$

Correspondingly, $x \in [i_t + [t] - L : i_t]$ implies

$$x \leq i_t. \tag{2}$$

If we subtract Eq. (2) from Eq. (1), we obtain

$$L \geq i_s + [s] - i_t.$$

Since $i_s \geq i_t$, we particularly have $L \geq |i_s + [s] - i_t|$. Furthermore, since we picked s and t arbitrarily, this proves that condition (O2) is satisfied. The proof for the second case $i_t > i_s$ works correspondingly with the roles of s and t exchanged.
(2)⇒(1): Let A be an L-occurence 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. □

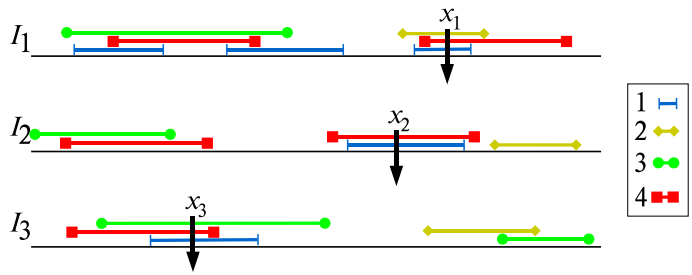

FIGURE 1. Example of an $(\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3)$-barbeque $A = \{1, 4\}$, which also is a best barbeque.

Given a set of fragments and a genome, we are particularly interested in L-occurences 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 barbeque ingredients instead of a color to each interval and identify the string T with a barbeque plate. Then, in order to have a tasty barbeque, our goal is to stab as many different features as possible with a skewer by stabbing only once into the plate. If only one barbeque plate is involved, this constitutes the *single person Best Barbeque Problem*. Before we generalize this problem to more than one barbeque plate, we sketch a simple algorithm for the single person Best Barbeque Problem.

The algorithm is based on sweeping the arrangement $\mathcal{I}$ of colored intervals. We call an interval C a *cell w.r.t.* $\mathcal{C}$ if each $x \in C$ stabs the same set of colors and every interval $C' \supset C$ contains at least one $x' \in C'$ that stabs a different set of colors than some $x \in C$. Each cell C of the arrangement induced by $\mathcal{I}$ is uniquely associated with a set of colors A_C – namely, we have $j \in A_C$ iff the cell C is covered by an interval of color j. (In fact, we will often refer to a cell simply as a set of colors). While sweeping over the interval arrangement, we can maintain the number and the set of all colors active in the current cell together with the maximum number of active colors that has been encountered in any of the preceding cells. Since between two neighbored cells, at most m changes between the corresponding sets can occur, all this can be achieved in $O(m)$ time using a boolean array of length m. Moreover, the arrangement contains at most $N := [T]$ many cells and sorting the interval borders takes $O(N \log N)$ time, so that we obtain an overall running time of $O(N(m + \log N))$.

4. The Best Barbeque Problem

The Best Barbeque Problem becomes a much more delicate problem if more than one barbeque plate is involved. The idea behind the generalization to K barbeque plates is as follows: suppose we have K guests invited to a barbeque, for each of whom we have prepared one plate with a selection of our m different barbeque 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 barbeque 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 every skewer, some skewers may contain additional features. For an example of the formal definition below, see Fig. 1.

Definition 3. *Let $\mathcal{I}_1, \ldots, \mathcal{I}_K$ denote K sets of intervals, each interval being assigned a color $j \in [1:m]$. We say that a set $A \subseteq [1:m]$ is an* $(\mathcal{I}_1, \ldots, \mathcal{I}_K)$-barbeque *if for each $i \in [1:K]$, there is an integer x_i such that for each color $a \in A$, x_i stabs at least one interval of color a in $\mathcal{I}_i$.*

A barbeque of maximum cardinality will also be referred to as a best barbeque 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 4. Instance: *Integers m, K; $\mathcal{I}_1, \ldots, \mathcal{I}_K$ denote K sets of intervals, each interval being assigned a color $j \in [1:m]$.*
Best Barbeque Problem (BBQ)**:** *What is the best barbeque of $\mathcal{I}_1, \ldots, \mathcal{I}_K$?*
Barbeque Decision Problem (DBBQ)**:** *Given an integer θ, is there an $(\mathcal{I}_1, \ldots, \mathcal{I}_K)$-barbeque whose cardinality is at least θ?*

Before we turn to the computational complexity of finding best barbeques, we explain its biological relevance using the equivalence of arrangements of colored intervals and L-occurences stated in Lemma 2: this equivalence tells us that a best barbeque in K sequences corresponds to a clustered L-occurence that *simultaneously* occurs in K genomes. This is in fact what biologists want to find out: whether there is an L-occurence that simultaneously occurs in several genomes and, in addition, involves a significant number of the candidate fragments s_j, then it is very likely that this clustered occurence constitutes a functionally relevant region. Hence, the fragments involved can be identified as being functionally responsible for some trait shared by the species corresponding to the K genomes.

Beyond this biological application, note that the definition of the Best Barbeque Problem naturally generalizes to colored arrangements of arbitrary geometric objects (such as discs or balls in higher dimension) rather than intervals in one dimension.

4.1. **Combinatorial Barbeques.** 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 [12]).

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 cells induced by $\mathcal{I}_i$. Corresponding to the geometric setting, we say that a set A is a $(\mathcal{C}_1, \ldots, \mathcal{C}_K)$-barbeque iff for each $i \in [1:m]$, there is a $C_i \in \mathcal{C}_i$ such that $A \subseteq C_i$. It is easy to see that every $(\mathcal{I}_1, \ldots, \mathcal{I}_K)$-barbeque is a $(\mathcal{C}_1, \ldots, \mathcal{C}_K)$-barbeque and vice versa.

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

Problem 5. 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 $\mathcal{C}_i = \{C_{i,1}, \ldots, C_{i,\lambda_i}\}$.*
Combinatorial Best Barbeque Problem (CBBQ)**:** *Maximize*

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

with $(\nu_1, \ldots, \nu_K) \in [1:\lambda_1] \times \cdots \times [1:\lambda_K]$.
Combinatorial Barbeque Decision Problem (DCBBQ)**:** *Given an integer θ, determine whether there are integers $(\nu_1, \ldots, \nu_K) \in [1:\lambda_1] \times \cdots \times [1:\lambda_K]$ such that*

$$|\cap_{i\in[1:K]} C_{i,\nu_i}| \geq \theta.$$

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 $|\cap_{i\in[1:K]} C_{i,\nu_i}|$, and keep track of the vector $(\tilde{\nu}_1, \ldots, \tilde{\nu}_K)$ that yields the largest cardinality intersection.
(A2) Enumerate all subsets of $[1:m]$. For each $A \subseteq [1:m]$, check whether there are suitable indices $\nu_1, \ldots, \nu_K$ such that $A \subseteq \cap_{i\in[1:K]} C_{i,\nu_i}$. 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, so that there is little hope to find a polynomial time algorithm. However, since the problem is of practical relevance, we provide branch-and-bound approaches in Section 5.2, 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 6.3. Finally, we provide an algorithm that is exponential in another, rather subtly hidden parameter, which is done in Section 6.

5. Complexity and Algorithms

5.1. **NP-completeness Results.** Our goal in this section is to prove the following:

Theorem 6.

(1) *DCBBQ is NP-complete.*
(2) *DBBQ is NP-complete.*

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

Our reduction from a K-clique in a K-partite graph to CBBQ will start with the problem of deciding whether there is a K-clique in a K-partite graph. Let $G = (V, E)$ denote an undirected K-partite graph, i.e., we have $V = V_1 \cup \dots \cup V_K$ 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). As has been noted by several authors and formally proved by Azarenok *et al.* in [2], deciding whether G has a K-clique is NP-complete.

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

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

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:

Lemma 7. *Using the notation introduced above, let $v_1 \in V_1, \dots, v_K \in V_K$. The following holds:*

(1) $\{u, v\} \in E \iff \{u, v\} \subseteq C_u \cap C_v$,
(2) $\cap_{i\in[1:K]} C_{v_i} \subseteq \{v_1, \dots, v_K\}$,
(3) $|\cap_{i\in[1:K]} C_{v_i}| = K \iff G$ *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 $u \in C_v$. This proves $u \in C_u \cap C_v$. The proof for $v \in C_u \cap C_v$ works analogously, 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 $x \in \cap_{i\in[1:K]} C_{v_i}$, and assume that $x \notin \{v_1, \dots, v_K\}$. 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 $|\cap_{i\in[1:K]} C_{v_i}| = K$. Then part (2) of this Lemma implies that $\cap_{i\in[1:K]} C_{v_i} = \{v_1, \dots, v_K\}$. 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 $\{v_i, v_j\}$.
Conversely, let $\{v_1, \dots, v_K\}$ 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 $\{v_1, \dots, v_K\} \subseteq \cap_{i\in[1:K]} C_{v_i}$, implying $|\cap_{i\in[1:K]} C_{v_i}| \geq K$. Claim (2) of this Lemma immediately implies $|\cap_{i\in[1:K]} C_{v_i}| \leq K$, so that we have $|\cap_{i\in[1:K]} C_{v_i}| = K$. □

Proof of Theorem 6. We start with the *proof of (1).*

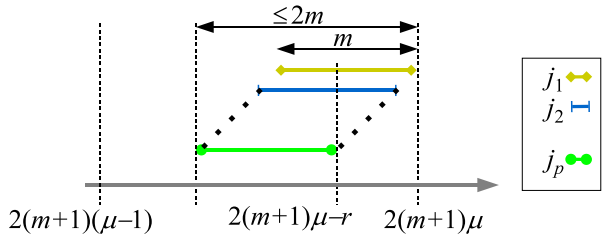

FIGURE 2. Constructing a set of colored intervals from a set $C_\mu = \{j_1, \ldots, j_p\} \in \mathcal{C}_i$.

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 barbeque decision problem, part (3) of Lemma 7 reduces the decision problem whether a K-partite graph has a K-clique to the combinatorial barbeque decision problem. Since the construction can be performed in polynomial time, this immediately yields the desired NP-completeness proof.

The *proof of (2)* is rather technical and hence we only provide a sketch. We provide a reduction from DCBBQ to DBBQ that essentially works as follows: given an instance of DCBBQ with sets $\mathcal{C}_1, \ldots, \mathcal{C}_K$ of subsets of $[1 : m]$, we construct an instance of DBBQ of intervals $\mathcal{I}_1, \ldots, \mathcal{I}_K$ colored with $[1 : m]$ that has exactly the same best barbeque. To this end, observe that we can add certain sets to each $\mathcal{C}_i$ without changing the best barbeque: if we have a set M satisfying $M \subseteq C$ for some $C \in \mathcal{C}_i$, then $\mathcal{C}_1, \ldots, \mathcal{C}_i \cup \{M\}, \ldots, \mathcal{C}_K$ obviously has the same best barbeque as $\mathcal{C}_1, \ldots, \mathcal{C}_K$.

We now construct $\mathcal{I}_i$ from $\mathcal{C}_i$ as follows: let $\mathcal{C}_i = \{C_1, \ldots, C_\lambda\}$. For each $\mu \in [1 : \lambda]$, we introduce $|C_\mu| =: p$ many intervals, in a way such that none of the intervals constructed from C_μ intersects with an interval constructed from any $C' \in \mathcal{C}_i$ other than C_μ. The crucial idea is that the intervals constructed from C_μ induce a cell that is equal to C_μ, beside a limited number of cells that are subsets of C_μ (which, as noted above, do not change the best barbeque).

Writing $C_\mu = \{j_0, \ldots, j_{p-1}\}$, we construct $|C_\mu| =: p$ colored intervals of length m from C_μ, namely for each $r \in [0 : p-1]$ we introduce the interval $[2(m+1)\mu - m + 1 - r : 2(m+1)\mu - r]$ colored with j_r (see Fig. 2 for an illustration). This way, we obtain an interval arrangement where $2(m+1)\mu - r$ stabs C_μ. Note that all other cells induced by the p intervals we constructed from C_μ are subsets of C_μ and hence do not change the best barbeque.

Altogether, the interval arrangements $\mathcal{I}_1, \ldots, \mathcal{I}_K$ can be shown to satisfy the following properties:

(i) For each $i \in [1 : K]$ and for each $C \in \mathcal{C}_i$, there is a cell induced by $\mathcal{I}_i$ that equals C.
(ii) For each $i \in [1 : K]$ and for each cell A induced by $\mathcal{I}_i$, there is a $C \in \mathcal{C}_i$ such that $A \subseteq C$.

Based on these two properties, in turn, one can show that the combinatorial best barbeque of $\mathcal{C}_1, \ldots, \mathcal{C}_K$ is equal to the geometric best barbeque of $\mathcal{I}_1, \ldots, \mathcal{I}_K$. □

Note that the above reduction only needs a slight modification to obtain a reduction from DCBBQ to finding maximum L-occurences: take $\Sigma := \{\mathrm{O}, \alpha_1, \ldots, \alpha_m\}$, and, rather than introducing an interval $[2(m+1)\mu - m + 1 - r : 2(m+1)\mu - r]$in $\mathcal{I}_i$, we place character α_r at position $2(m+1)\mu - r$ in the string T_i. T_i is constructed as a string of length $2(m+1)|C_\mu|$. All positions in T_i that are not filled with a character α_r by the above construction will be filled with O. This already yields the reduction, as can be seen from Theorem 6 using Lemma 2.

5.2. **Branch-and-Bound Algorithms.** Studying the algorithm specified in the last paragraph of Section 4.1 in more detail, one realizes that the branch-and-bound principle can be applied as follows: Suppose we have already found a vector $(\tilde{\nu}_1, \ldots, \tilde{\nu}_K)$ such that $\cap_{i \in [1:K]} C_{i,\tilde{\nu}_i} = \theta$. 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 $\cap_{i \in [1:a]} C_{i,\nu_i} \leq \theta$, we know

that no matter how we choose $\nu_{a+1}, \ldots, \nu_K$, the cardinality of the intersection $\cap_{i\in[1:K]} C_{i,\nu_i}$ cannot exceed θ. In terms of a branch-and-bound algorithm, this means that if t denotes the cardinality of the best barbeque so far, then $|\cap_{i\in[1:a]} C_{i,\nu_i}| \leq t$ 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.

As can be easily seen, 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. To this end, observe that if some $A \subseteq [1 : m]$ is not an $(\mathcal{I}_1, \ldots, \mathcal{I}_K)$-barbeque, then all sets A' with $A \subseteq A'$ are not barbeques either. In particular, sets that are not barbeques cannot be best barbeques. In terms of a branch-and-bound algorithm, this means that if we encounter a set A that is not a barbeque, 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 superset of 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 $O(2^m \Lambda m)$.

6. Variants of the Best Barbeque Problem

6.1. **Bounded Differences.** In the barbeque-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 be disturbed.

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

Suppose we are given an instance of the combinatorial best barbeque problem, together with a barbeque $B = \cap_i C_{i,\nu_i}$. The number of "extra" features occuring in C_{i,ν_i} now reads as $|C_{i,\nu_i} \setminus B|$. Correspondingly, we say that B is a *δ-bounded barbeque* if there are indices $\nu_1, \ldots, \nu_K$ sucht 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 barbeque. To this end, 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 barbeque. 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 barbeque is easy: Taking $\mathcal{C} := \cup_i \mathcal{C}_i$, we test for each $B \in \mathcal{C}$ 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 $B \in \mathcal{C}$

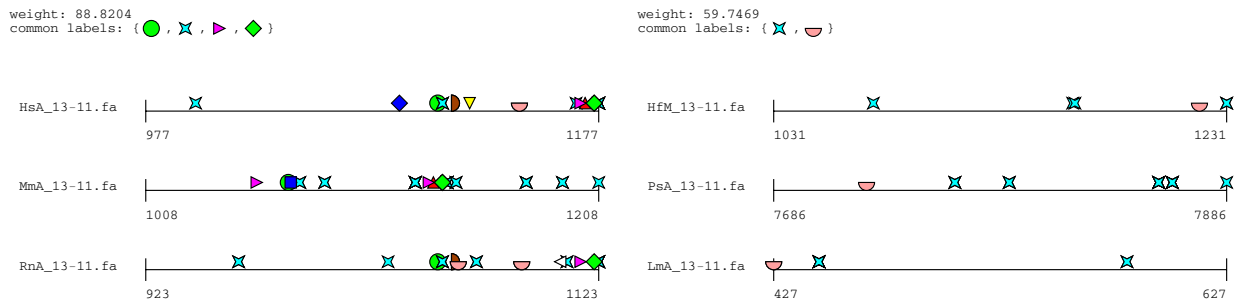

FIGURE 3. A significant cluster of binding sites among evolutionary closely related species (Human, Mouse and Rat) (left). Searching for the same set of candidate binding sites in evolutionary more unrelated species, one obtains a smaller and hence less significant – probably non-functional – cluster (right). Both clusters were obtained with the implementation of a weighted version of Algorithm (A2).

yields the largest cardinality 0-bounded barbeque. Since we test $|\mathcal{C}| = \Lambda$ many sets B whether B is a 0-bounded barbeque, so that the overall running time amounts to $O(\Lambda K m \log \Lambda)$.

This idea carries to finding largest cardinality δ-bounded barbeques. To this end, each of the sets $\mathcal{C}_i$ needs to be supplemented as follows:

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

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 barbeques now works the same way as the algorithm for 0-bounded barbeques, 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 $O(m^\delta \Lambda K m \delta \log(m\Lambda))$.

6.2. **Weighted Versions.** In this section, we provide a basis to find maximum *weighted* barbeques. To this end, given a (finite) set M, 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 above carry naturally to the weighted version of the best barbeque 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.

6.3. **Results and Perspectives.** As a first and simple example for the application of the BBQ approach to biological data we consider a particular footprint cluster in the intergenic regions between the *HoxA13* and *HoxA11* genes in vertebrates, which have a length between 12000 and 15000 nucleotides. *Hox* genes are a class of transcription factors that have a crucial role in early embryonic development [9]. They appear in tightly linked gene clusters. We used the region between *HoxA13* and *HoxA11* in $K = 3$ different species as our genomic sequences, and a set of $m = 15$ candidate binding sites obtained with the tool `tfsearch` [1] on a preselected region of the intergenic region under consideration, which is supposed to play a role in limb development. See Fig. 3 for a visualization and brief discussion of the results.

Obtaining the above results (with $K = 3$, $m = 15$ and genomic sequences with a length between 12000 and 15000) with an implementation of Algorithm (A2) took few seconds of computation time on a standard desktop computer with a 2.8 MHz processor. Using the bounded difference method from Section 6 and using $\delta = 2$ as a bound, even instances with $K = 5$ and $m = 300$ can be computed within below one minute.

Determining L-occurences as described in this work provide means by which one may discover structural or spatial features shared by several objects. Looking at Algorithm (A1), the complexity of the Best Barbeque Problem is somewhat related to finding longest common subsequences [8]: If the number K of sequences – in our case interval arrangements – is fixed, then there is an $O(N^K)$ time algorithm; if K is an input variable, the problem is NP-complete. Since computing certain weighted variants of longest common subsequences – multiple sequence alignments – is a very common and relevant task in computational biology, numerous approximation algorithms have been developed for this task.

Beside the analogy in computational complexity, it is also important to note the conceptual similarity between multiple sequence alignments and best barbeques: the reason why multiple sequence alignments take an extremely important place in biological applications is that simultaneous comparison of many sequences allows one to find similarities that are invisible in pairwise comparisons. Now, sequence alignments are always based on finding order preserving mappings between regions of sequences, so that the concept of sequence alignments cannot be carried into non-sequential objects. The key difference between multiple sequence alignments and the best barbeque approach is that the Best Barbeque Problem does not involve a linear order on the objects. This allows a generalization of a multiple-alignment-like concept to non-sequential structures and objects – as long as in some way, features are available as the basis for obtaining colored arrangements.

7. Acknowledgments

We acknowledge Andreas Dress for helpful remarks. Furthermore, computational assistance by Manja Lindemeyer is gratefully acknowledged. This work was supported in part by the *DFG* Bioinformatics Initiative BIZ-6/1-2.

References

[1] Yutaka Akiyama. Tfsearch: Searching transcription factor binding sites, 1998.

[2] A.S. Azarenok and V.S. Krikun. A clique in an 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.

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

[4] L. Duret and P. Bucher. Searching for regulatory elements in human noncoding sequences. *Curr. Opin. Struct. Biol.*, 7:399–406, 1997.

[5] J.W. Fickett and W.W. Wasserman. Discovery and modeling of transcriptional regulatory regions. *Current Opinion in Biotech.*, 11:19–24, 2000.

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

[7] 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.

[8] D. Maier. The complexity of some problems of subsequences and supersequences. *Journal of the ACM*, 25(2):322–336, 1978.

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

[10] Pavel A. Pevzner. *Computational Molecular Biology*. MIT Press, 2000.

[11] Roded Sharan, Asa Ben-Hur, Gabriela G. Loots, and Ivan Ovcharenko. CREME: Cis-regulatory module explorer for the human genome. *Nucl. Ac. Res.*, 32:W253–W256, 2004.

[12] M. Sharir and P.K. 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.

[13] D.A. Tagle, B.F. Koop, M. Goodman, J.L. Slightom, D.L. Hess, and R.T. Jones. Embryonic ϵ and γ globin genes of a prosimian primate (galago crassicaudatus). nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. *J. Mol. Biol.*, 203(2):439–455, 1988.

[14] D Tautz. Evolution of transcriptional regulation. *Curr. Opin. Genet. Dev.*, 10:575–579, 2000.