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 two branch-and-bound algorithms to detect footprint clusters in some real world instances of phylogenetic footprints. Finally, we point out other geometric and combinatorial scenarios where optimizing barbeques might be of practical relevance.

1. Introduction

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 [2]. 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* [12, 6]. It has been noted for a long time, however, that non-coding sequences can contain islands of strongly conserved segments, so-called *phylogenetic footprints* [11]. In many cases phylogenetic footprints have experimentally been shown to be indicative of functional cis-regulatory elements, see e.g. the reviews [3, 4].

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 sequences fragments that occur clustered (i.e., close to each other) on several large genomic sequence fragments.

In this paper, we introduce a computational geometry based approach to this question, which is based on stabbing colored intervals from a certain arrangement of colored intervals partitioning a genome into equivalence classes. The outline of this paper is as follows: In the following section, we give a formal problem description, and provide the basic ideas of the biological relevance of the problems 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 can be rephrased as a combinatorial optimization problem. In Section 4.1 we show that this combinatorial version of the Best Barbeque Problem is NP-complete. We then provide two branch-and-bound 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 useful for different types of instances. Finally, we discuss a further geometric generalization that indicates other potential applications of algorithms for the Best Barbeque Problem.

2. 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 $a < b$. 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, $S|_{a,b}$ the substring $\sigma_a \sigma_{a+1} \ldots \sigma_b$. We say that a string U *occurs in T at position x* iff $T|_{x,x+[U]-1} = U$. 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 $s_1, \ldots, s_m, T \in \Sigma^*$, $L \in \mathbb{N}$ and $A = \{a_1, \ldots, a_\ell\} \subseteq [1 : m]$ with $|A| =: \ell$. We say that A* is an L-occurence in T w.r.t. $s_1, \ldots, s_m$ *if there are indices $i_1, \ldots, i_\ell$ such that*

(O1) *s_{a_j} occurs in T at position i_j and*
(O2) *$|i_\nu + [s_{a_\nu}] - i_\mu| \leq L$ for all $\mu, \nu \in [1 : \ell]$.*

Correspondingly, we refer to $A \subseteq [1 : m]$ together with indices $\langle i_1, \ldots, i_\ell \rangle$ satisfying the above conditions as an L-occurence *of $s_{a_1}, \ldots, s_{a_\ell}$ in T.*

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 3 – 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 $s_1, \ldots, s_k$ as in Definition 1, we obtain a set of colored intervals in the following way: first, identify each fragment s_j with a color j. Now, introduce an interval $[p + [s_j] - L : p]$ with color j whenever s_j occurs at position p in T. We will also refer to the set of colored intervals

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

as *the set of intervals induced by $s_1, \ldots, s_m$ 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 $s_1, \ldots, s_m$ in T with cluster length L. Furthermore, let $A \subseteq [1 : m]$. Then, the following statements are equivalent:*

(1) *There is an integer x such that for all $a \in A$, x stabs an interval in I with color a.*
(2) *A is an L-occurence in T w.r.t. $s_1, \ldots, s_m$.*

Proof. Let $A = \{a_1, \ldots, a_\ell\}$.
(1)⇒(2): Since x stabs one interval of each color contained in A, x is contained in at least ℓ intervals $[h_1 : i_1], \ldots, [h_\ell : i_\ell]$, each with color $a_1, \ldots, a_\ell$, respectively. Note that by construction of I, we have $h_\mu = i_\mu + [s_{a_\mu}] - L$ for $\mu \in [1 : \ell]$. Since, by construction of I, s_{a_μ} occurs at position i_μ for each $\mu \in [1 : \ell]$, condition (O1) of an L-occurence is satisfied, and it remains to prove that condition (O2) holds.

Without loss of generality, let $i_1 \leq \cdots \leq i_\ell$. Due to $x \in [h_1 : i_1]$, we particularly have

$$x \leq i_1 \leq \cdots \leq i_\ell.$$

Now, pick $\mu, \nu \in [1 : \ell]$ arbitrarily. We distinguish two cases, starting with $i_\nu \leq i_\mu$. Then $x \in [i_\mu + [s_{a_\mu}] - L : i_m u]$ implies

$$x \geq i_\mu + [s_{a_\mu}] - L. \tag{1}$$

Correspondingly, $x \in [i_\nu + [s_{a_\nu}] - L : i_\nu]$ implies

$$x \leq i_\nu. \tag{2}$$

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

$$L \geq i_\mu + [s_{a_\mu}] - i_\nu.$$

Since $i_\mu \geq i_\nu$, we particularly have $L \geq |i_\mu + [s_{a_\mu}] - i_\nu|$. Furthermore, since we picked μ and ν arbitrarily, this proves that condition (O2) is satisfied. The proof for the second case $i_\nu > i_\mu$ works correspondingly with the roles of μ and ν exchanged.

(2)⇒(1): Let A be an L-occurence in T. Then, by condition (O1), there are indices $i_1, \ldots, i_\ell$ such that s_{a_μ} occurs at position i_μ in T. Without loss of generality, let

$$i_1 \leq \cdots \leq i_\ell. \tag{3}$$

Then, applying (O2), we get

$$|i_\mu + [s_{a_\mu}] - i_1| \leq L.$$

Dropping the absolute value due to $i_1 \leq i_\mu$, we get $i_1 \geq i_\mu + [s_{a_\mu}] - L$. Together with Eq. (3), this yields $a \in [i_\mu + [s_{a_\mu}] - L : i_\mu]$ for all $\mu \in [1 : \ell]$. Since for each μ, this interval is contained in I with color a_μ, we are done. □

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 in sweeping the arrangement I of colored intervals. Each cell C of the arrangement induced by 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. 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, so that we obtain an overall running time of $O(Nm)$.

3. 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.

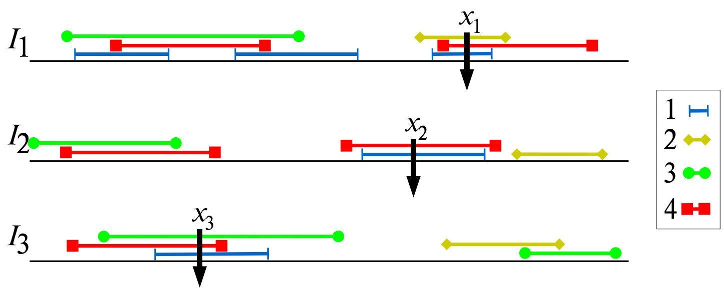

FIGURE 1. Example of am (I_1, I_2, I_3)-barbeque $A = \{1, 4\}$, which also is a best barbeque.

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 (BBQ)**.** *Let $I_1, \ldots, I_K$ 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 $(I_1, \ldots, I_K)$-*barbeque *if for each $i \in [1 : K]$, there is an integer x_i such that for each color $a \in A$, x_i that stabs at least one interval of color a in I_i.*

A barbeque of maximum cardinality will also be referred to as a best barbeque of $I_1, \ldots, I_K$.

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

Problem 4. Instance: *Integers m, K; $I_1, \ldots, I_K$ 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 $I_1, \ldots, I_K$?*
Barbeque Decision Problem (DBBQ)**:** *Given an integer θ, is there an $(I_1, \ldots, I_K)$-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: if 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 a being functionally responsible for some trait shared by the species corresponding to the K genomes.

Note that the definition of the Best Barbeque Problem naturally generalizes to arrangements of other geometric objects than just intervals in one dimension. Some further remarks of such generalizations are given in Section 5.2.

3.1. **Combinatorial Barbeques.** Given a set of colored intervals I, we canonically obtain an equivalence relation between integers – each integer x stabs a certain set of colors in I; we define $x \sim y$ (w.r.t. I) iff x stabs the same set of colors in 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 [10]).

Given K sets of colored intervals, the cells induced by each 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 $C_1, \ldots, C_K$, where C_i denotes the cells induced by I_i. Corresponding to the geometric setting, we say that a set A is a $(C_1, \ldots, C_K)$-barbeque iff for each $i \in [1:m]$, there is a $B_i \in C_i$ such that $A \subseteq B_i$. It is easy to see that every $(I_1, \ldots, I_K)$-barbeque is a $(C_1, \ldots, C_K)$-barbeque and vice versa.

Hence, computing the induced cells for each I_i leaves us with the following problem:

Problem 5. Instance: *Integers m, K; $C_1, \ldots, C_K$ denoting K sets of subsets of $[1:m]$, with $\lambda_i := |C_i|$ and $C_i = \{B_{i,1}, \ldots, B_{i,\lambda_i}\}$.*
Combinatorial Best Barbeque Problem (CBBQ)**:** *Maximize*

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

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*

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

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 $|\cap_{i\in[1:K]} B_{i,\nu_i}|$, 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 $A \subseteq \cap_{i\in[1:K]} B_{i,\nu_i}$. 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 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 5.3.

4. Complexity and Algorithms

4.1. DCBBQ is NP-complete.

Our goal in this section is to prove the following:

Theorem 6. *The combinatorial barbeque decision problem 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 $|\cap_i B_{i,\nu_i}|$ in $O(mK)$ time.

Note that the combinatorial barbeque decision problem easily reduces to DBBQ, so that the above theorem (in combination with an analogous argument showing that DBBQ is in NP) implies the NP-completeness of DBBQ as well. Also, given an instance of DCBBQ, it is easy to construct an instance of a simultaneous L-occurence having the same solution, so that our original and biologically relevant problem is NP-complete as well.

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 $V = V_1 \cup \cdots \cup V_K$ 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). As has been noted by several authors and formally proved by Azarenok *et al.* in [1], deciding whether G has a K-clique is NP-complete.

Given a K-partite graph G, we now construct a collection $C_1, \ldots, C_K$ 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 $B_v := N(v) \cup \{v\}$. The following Lemma establishes close connections between the graph G and intersections of the sets B_v:

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

(1) $\{u, v\} \in E \iff \{u, v\} \subseteq B_u \cap B_v$,
(2) $\cap_{i \in [1:K]} B_{v_i} \subseteq \{v_1, \ldots, v_K\}$,
(3) $| \cap_{i \in [1:K]} B_{v_i}| = K \iff$ *G has a K-clique.*

Proof. (1): Let $\{u, v\} \in E$. Then, by construction, we have $u \in B_u$ and $u \in N(v)$, and hence also $u \in B_v$. This proves $u \in B_u \cap B_v$. The proof for $v \in B_u \cap B_v$ works analogously, so that we have $\{u, v\} \subseteq B_u \cap B_v$.
Conversely, let $\{u, v\} \subseteq B_u \cap B_v$. Then, $v \in B_u$ implies $v \in N(u)$, and hence $\{u, v\} \in E$.
(2): Let $x \in \cap_{i \in [1:K]} B_{v_i}$, 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 $x \in B_{v_1}$. Now, by construction, the only vertex from V_1 contained in B_{v_1} is v_1 itself. However, we assumed that $v_1 \neq x \in B_{v_1}$, which is a contradiction.
(3): Let $|\cap_{i \in [1:K]} B_{v_i}| = K$. Then part (2) of this Lemma implies that $\cap_{i \in [1:K]} B_{v_i} = \{v_1, \ldots, v_K\}$. 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 $\{v_i, v_j\} \in B_{v_1} \cap B_{v_2}$. Using part (1) of this Lemma, this implies $\{v_i, v_j\}$.
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 $v_i \in B_{v_j}$. By construction, we also have $v_i \in B_{v_i}$. Altogether, we obtain $\{v_1, \ldots, v_K\} \subseteq \cap_{i \in [1:K]} B_{v_i}$, implying $| \cap_{i \in [1:K]} B_{v_i}| \geq K$. Claim (2) of this Lemma immediately implies $| \cap_{i \in [1:K]} B_{v_i}| \leq K$, so that we have $| \cap_{i \in [1:K]} B_{v_i}| = K$. □

Proof of Theorem 6. Since choosing $C_i := \{B_{v_i} \mid v_i \in V_i\}$ 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. This immediately yields the desired NP-completeness proof. □

4.2. **Branch-and-Bound Algorithms.** Studying the algorithm specified in the last paragraph of Section 3.1 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 $\cap_{i \in [1:K]} B_{i,\tilde{\nu}_i} = \theta$. 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 $\cap_{i \in [1:a]} B_{i,\nu_i} \leq \theta$, we know that no matter how we choose $\nu_{a+1}, \ldots, \nu_K$, the cardinality of the intersection $\cap_{i \in [1:K]} B_{i,\nu_i}$ 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 $|\cap_{i \in [1:a]} \mid B_{i,\nu_i}| \leq t$ 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 $(I_1, \ldots, I_K)$-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 C_i. Finally, it is easy to see that, with $\Lambda := |C_1| + \cdots + |C_K|$, the running time of Algorithm (A2) is $O(2^m \Lambda m)$.

5. Variants of the Best Barbeque Problem

5.1. **Weighted Versions.** In this section, we provide an approach to find maximum *weighted* barbeques. We limit our considerations to weighted versions of the combinatorial version of the Best Barbeque Problem. These carry immediately into weighted versions of the geometric version and, in particular, to weighted L-occurences that were our starting point. Weighted L-occurences are highly relevant for detecting phylogenetic footprint clusters: often, the binding site fragments s_j do not occur exactly within the footprint cluster, but with a (limited) number of mismatches. Correspondingly, the higher the number of mismatches of an occurence, the lower the weight associated with the occurence should be. This carries immediately to weights for the intervals induced by $s_1, \ldots, s_m$ and hence to weighted subsets for the cells obtained from these intervals.

Moreover, longer binding site fragments can be ranked higher by assigning a length-dependent weight to each s_j. (After all, in the barbeque illustration of the problem, we want to rank large pieces of ingredients higher than small pieces.) Dealing with weighted sets, we now need to supply a suitable concepts to handle these. In the face of both the definition of the combinatorial Best Barbeque Problem and Algorithm (A1) being based on set intersections, we provide a suitable generalization of set intersections to weighted set intersections.

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

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

Note that this operation can easily be seen to be associative. Moreover, define the weight of a mapping $X\colon M \to \mathbb{R}_{\geq 0}$ as $w(X) := \sum_{a \in M} X(a)$. Using these notions of weighted subsets, the specification of Problem 5 immediately yields a weighted version of the Best Barbeque Problem: we now have a collection of K sets of weighted subsets of $[1 : m]$, $C_1, \ldots, C_K$, and we say that a weighted subset A of $[1 : m]$ is a barbeque iff there are weighted subsets $B_1 \in C_1, \ldots, B_K \in C_K$ such that $A = \cap_i B_i$. The best weighted barbeque of $C_1, \ldots, C_K$ then is the barbeque with maximum weight. As one can easily see, this maximum can still be found using Algorithm (A1) which checks all possible combinations of weighted subsets from the different C_i.

5.2. **Generalized L-occurences.** As already noted above, the Best Barbeque Problem generalizes canonically to arrangements of other geometric objects than just one dimensional intervals. A reasonable question that arises is to ask whether there are also corresponding generalizations of L-occurences and Lemma 2 to higher dimensions. In order to at least partially answer this, suppose we are given K collections of colored points $T_1, \ldots, T_K$ in a metric space X with distance function $d\colon X \times X \to \mathbb{R}_{\geq 0}$ (such as the two dimensional plain with Euclidean distance), each point being attributed with a color (where in practice, each color might correspond to a certain feature). I.e., each T_i is a subset of $X \times [1 : m]$. Given $L \in \mathbb{R}$ and $A = \{a_1, \ldots, a_\ell\} \subseteq [1 : m]$, we can now say that A is an L-occurence if, in analogy to Definition 1, there are points $p_1, \ldots, p_\ell$ satisfying the following conditions:

(O1$'$) $(p_j, j) \in T$ and
(O2$'$) $d(p_\nu, p_\mu) \leq L$ for all $\mu, \nu \in [1 : \ell]$.

Choosing $X := \mathbb{R}$ corresponds to substituting the genome string T with the real line and instead of occurences of some s_j, we simply have points on the real line with color j (such that a point with color j conceptually corresponds to an occurence of a fragment s_j of length 0). Now, instead of assigning the interval $[x - L, L]$, we identify $(x, j) \in T$ with the interval $[x - L/2, x + L/2]$, i.e., the closed $L/2$-neighborhood of x. (Note that this corresponds to shifting all intervals by

	Hf	Ps	Lm	Xt	Gg	Hs	Mm	Rn
Hf	-							
Ps	CES	-						
Lm	*	*	-					
Xt	*	(CGS)	*	-				
Gg	CES	(CGS)	*	CEGSX	-			
Hs	CES	(CGS)	*	CEGPSX	CEGMSX	-		
Mm	*	*	*	CEGX	CEGX	BCEGX	-	
Rn	*	(CNPS)	*	CGPSX	CGSX	BCGPSX	BCGX	-

B *Brn-2*, C *CdxA*, E *Evi-1*, G *GATA-1*, M *v-Myb*, N *Nkx-2*, P *Pbx-1*, S *SRY*, X *XFD-1*.

TABLE 1. BBQ solution for all pairwise comparisons in the *Hox* example with $L = 200$ and $m = 15$ and no mismatches. A $*$ indicates solutions outside the footprint cluster displayed in Fig. 2, solutions in parenthesis match this cluster in one sequence, while the position of the solution interval in bichir sequence, which does not contain the footprint cluster according to the `tracker` results, is located some 8kb downstream of the cluster.

$L/2$ and does not change the resulting interval arrangement at all.) Given $T \subseteq X \times [1 : m]$ and letting $N_\varepsilon(x)$ denote the closed ε-neighborhood of x, we now obtain a set of neighborhoods induced by T with cluster size L, namely

$$I := \{(N_{L/2}(x), j) \mid (x, j) \in T\}.$$

If A denotes some subset of $[1 : m]$, a generalization of Lemma 2 to metric spaces would need to set up an equivalence between the following statements:

(1) There is an $x \in M$ such that for all $a \in A$, x stabs a neighborhoods in I with color a.
(2) A is an L-occurence in T.

Although (1) indeed implies (2), the other direction does not hold in general. However, at last a weaker version of the reverse direction can be shown to hold: If A is a $L/2$-occurence, then (1) follows. Hence, considering the Best Barbeque Problem on the arrangement of all $L/2$-neighborhoods at least gives an approximate solution to determining L-occurences in the generalized sense.

5.3. **Results and Perspectives.** As an example for the application of the BBQ approach to biological data we consider a particular footprint cluster in the intergenic region 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 [8]. They appear in tightly linked gene clusters. The `tracker` program [9] was used to search for phylogenetic footprints in the region between *HoxA13* and *HoxA11*, and a particular footprint cluster was singled out for further analysis, Fig. 2.

We solved the pairwise BBQP for all combinations of intergenic sequences using the sequence fragments shown in Figure 2. The results are summarized in Table 1. Generally, the solutions for comparisons of two tetrapod sequences contain more distinct putative binding sites and all recover the target cluster shown in Fig. 2. In contrast, comparisons of tetrapods with shark, bichir, or latimeria recover only a smaller number of motifs which, in addition, are among the smallest and hence most frequently found ones in the `transfac` database.

In the result above, we used an implementation of the branch and bound version of Algorithm (A1), which gives a significant speed up compared to a brute force implementation of Algorithm (A1). The main reason we computed a maximum L-occurences of two sequences only is that already for $K = 3$, the running times were unacceptably high. Each instance involved in the results above took few minutes of computation time on a state-of-the-art desktop computer. However, choosing $K > 2$ or m significantly larger than 15 rapidly increased the necessary computation time beyond several hours. Since in typical applications, one is interested in much larger instances (say, $K \geq 6$ and k in the order of up to several hundred), there clearly is

```
HfM,    949   ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
PsA,   1467   ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
LmA,    943   -AAAATTCAG GAATTGATCT TCGGTGCGGC TTTctttcat cacgtaacac gtgaatttac gttttattct gttattaact tggt------ ---------- ---------- ------ATTT AGTACGCTGT GGCAATATAT TTATtatgcg
XtA,    887   ---------- ---------- ---------- TTTTAATCAA A-ATAGATAT TTGAACAAat cattattttc ctggcTTATT CACTCAAAAA TGGTATCTTG TAAATAAt-- -----AAATT AGAGCACCGT GAACGTGTTT TTGTGCACaa
                                                --------->                      <------                                       -------------->
                                                    Pbx-1                       CdxA                                   <---------            CdxA
                                                   ------->                <-------------                              Evi-1         <--------------
                                                        SRY                Oct-1                                                     CdxA
                                                                                                                              ------------------------>
                                                                                                                       ======================>       S8
                                                                                                                 <=========              XFD-1
                                                                                                                 GATA-2
                                                                                                          <------->
                                                                                                          CdxA           <-------
                                                                                                       <----------------- CdxA
                                                                                                       YY1                 -------------->
                                                                                                                                     Oct-1
GgA,   1209   GGACGTGTAG GGATTCTTTT T-ACAGCACC TTTTAATCCA A-TCAGTTAT TTCAACCAGC ACATTa-TTT TGTTTTTATT CACTATAAGA AGCTATCGTG TAAATAAAag acaacAAATC AGCGCACTGT GAAAATGTAT TTGTGCAC--
                               <-----------               <-----------           <-------                        <=========               <----------                ------->
                               TATA                       v-Myb                  CdxA   <-------                 GATA-1/2                 GATA-1                         CdxA
                                                                                        SRY                            ==============>
                                                                                   <----------                                   XFD-1
                                                                                   Evi-1                                <-------
                                                                                    <-------------                      CdxA
                                                                                    SRY
HsA,   1081   GAAAGATTAG TGATTCATCT TCACAGCACA TTTTTAATCA A-GCAGTTAT TTCAACCAGC ACATTCGTTT TGTTCATATT CACTATAGAA TGATATCTTG TAAATAAAGA C-----ATTC AGCACACTGT GAAAATGTAT TTGTGCAC--
                       <-------->                                                                                  <---------                                         ------->
                       AP-1                                                                                         Evi-1                                                 CdxA
                                                 --------->                                                            ==============>
                                                      Pbx-1                        <------                    ------->           XFD-1
                                                          <----------              SRY                            CdxA
                                                          v-Myb                                                         <-------
                                          ------------------>                                                           CdxA
                                            <-------    Brn-2                                                    <=========   <------
                                            CdxA                                                                 GATA-2       CdxA

MmA,   1046   GAAAGTTTAG GGACTCATCG TCACAGCACA TTTTAAATCA A-GCAGTAAT TTCAGTTCAC ACATTCGTTC TGTTCATATT CACTGTAA-A TGCTATCTTG TAAATAAAGA C-----ATTC AGCACACTGT GAAAATGTAT TTGTGCAC--
                                                             <------                                               <---------                                         ------->
                                                             CdxA                                                  Evi-1                                                  CdxA
                                                <------                                                          <=========
                           <----------          CdxA                                                            GATA-1/2/3
                           GATA-1     <-----------------                                                <--------
                                           Tst-1                                                        CdxA          ==============>
                                          ------------------>                                                                    XFD-1
                                                        Brn-2                                                           <-------
                                                                                                                        CdxA <------
                                                                                                                             CdxA
RnA,   1027   GAAAATTTAG GGATTCATCT TCACAGCACA TTTTTAATCA AaGCAGTAAT TTCAATTCAC ACATTCGTTT TGTTCATATT CACTTTAA-A TGCTATCCTG TAAATAAAGA C-----ATTC AGCACACTGT GAAAATGTAC TTGTGCAC--
                                                             <------                                             <=========
                                                    -------> CdxA                                     <------    GATA1/2
                  ------->                               SRY                                       Nkx-2 <--------
                  CdxA                           --------->                        <------               CdxA
                                                      Pbx-1                        SRY                                  <-------
                                          ------------------>                                                          CdxA <------
                                            <-------   Brn-2                                                                 CdxA
                                            CdxA                                                                      ==============>
                                                                                                                                 XFD-1
```

FIGURE 2. Detailed view of a single footprint cluster in the intergenic region between *HoxA13* and *HoxA11* in different vertebrate species (Hf: horn shark (*Heterodontus francisci*); Ps: bichir (*Polypterus senegalus*), Lm: *Latimeria menadoensis*, Xt: claw frog (*Xenopus tropicalis*), Gg: chicken, Hs: human, Mm: mouse, Rn: rat). Below each sequence putative transcription factor binding sites are annotated using `tfsearch` by a comparison with the `transfac` database [5]. Note that transcription factor binding sites are detected only in the five Tetrapoda, while the three more basal vertebrates, shark, bichir (a primitive actinopterygian fish) and the coelacanth (a sacropterygian fish widely known as “living fossile”) do not share this particular footprint cluster.

a demand to develop better algorithms that can handle such instances. However, the results shown above also indicate that the approach is of high practical relevance.

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 [7]: 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. Being conceptually closely related to (local) multiple sequence alignments, some of the underlying ideas might carry into approximation algorithms for determining L-occurences and allow for computing larger instances of the problem.

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.

6. Acknowledgments

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] 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.

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

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

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

[5] 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. *Nucl. Acids Res.*, 26:364–370, 1998.

[6] 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.

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

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

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

[10] 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.

[11] 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.

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