Efficient Extraction of Structured Motifs Using Box-links

Alexandra M. Carvalho[1], Ana T. Freitas[1], Arlindo L. Oliveira[1], and Marie-France Sagot[2]

[1] IST/INESC-ID, Rua Alves Redol, 9, 1000-029 Lisboa, Portugal
{asmc, atf, aml}@algos.inesc-id.pt
[2] Inria Rhône-Alpes, Université Claude Bernard, Lyon I, 43 Bd du 11 Novembre 1918, 69622 Villeurbanne Cedex, France
Marie-France.Sagot@inria.fr

Abstract. In this paper we propose a new data structure for the efficient extraction of structured motifs from DNA sequences. A structured motif is defined as a collection of highly conserved motifs with pre-specified sizes and spacings between them. The new data structure, called box-link, stores the information on how to jump over the spacings which separate each motif in a structured motif. A factor tree, a variation of a suffix tree, endowed with box-links provide the means for the efficient extraction of structured motifs.

Structured motifs try to capture highly conserved complex regions in a set of DNA sequences which, in the case of sequences from co-regulated genes, model functional combinations of transcription factor binding sites [1,2,3]. Formally, a *motif* is a non-empty string over an alphabet Σ (e.g., Σ ={A,C,T,G} for DNA sequences). A *structured motif* [1] is a pair (m, d) where m is a p-tuple of motifs $(m_i)_{1\leq i\leq p}$, denoting p boxes, and d is a $(p-1)$-tuple of pairs $(d_{\min_i}, d_{\max_i})_{1\leq i<p}$, denoting $p-1$ intervals of distance. In the following, we consider that all p boxes of a structured motif have a fixed length k and a fixed distance between boxes d. The general case was studied but is out of the scope of this abstract. Algorithms and complexity results are easily adaptable to the more general case.

A *factor tree*, also called a *k-factor tree* [4], is a data structure that indexes the factors of a string whose length does not exceed k. In the following we define box-links, whose purpose is to store the information needed to jump from box to box in a structured motif, over a factor tree. Formally, let L be the set of leaves at depth k of a k-factor tree $\mathcal{T}$ for a string s of length n and L_k^i denote all possible i-tuples over L. A *box-link of size* i, with $1 \leq i < p$, is a $(i+1)$-tuple in L^{i+1} such that there is a substring s' of s where: (i) the length of s' is $ik+(i-1)d$; (ii) the k-length substring of s' ending at position $jk + (j-1)d$, with $1 \leq j \leq i$, is the path in $\mathcal{T}$ spelled from the root to the j-th leaf of the box-link tuple. Box-links can be used to extract structured motifs when built over a *generalized factor tree* (a factor tree for a set of N sequences). However, in this case, box-links have to be endowed with a $Colors$ Boolean array [1] in order to distinguish in which of the N input sequences the corresponding boxes are linked.

In the following, we present an algorithm to build box-links. The algorithm makes use of two variables. First, the variable $list_{leaf}$ has the list of all leaves inserted in the

factor tree, which can be easily obtained during the factor tree construction. In fact, for the sake of exposition, $list_{leaf}$ can be seen as a family of variables $(list_{leaf_i})_{1 \leq i \leq N}$, where each $list_{leaf_i}$ has average length n, the average length of an input sequence. Observe that the substring labeling the path from the root to the j-th leaf of $list_{leaf_i}$ corresponds to the j-th at most k-length substring of the i-th input string. Second, the variable b_j stores the j-size box-links being built. We now describe AddBoxLink function. AddBoxLink(b,v,i) adds a box-link between an existing $(j-1)$-size box-link b and a leaf v for the i-th input sequence. However, it only creates a new box-link if there is not already a box-link between box-link b and node v. In either way, creating or not a new box-link, the AddBoxLink function sets the Boolean array entry i to 1. The pseudo-code of the algorithm to build box-links is presented in Algorithm 1.

Algorithm 1 BoxLink(Boxes p, BoxSize k, BoxDistance d, ListLeaf $list_{leaf}$)

1. for i from 1 to N
2. while size of $list_{leaf_i} \geq pk + (p-1)d$
3. $b_0 =$ AddBoxLink$(nil, list_{leaf_i}[0], i)$
4. for j from 1 to $p-1$
5. $b_j =$ AddBoxLink$(b_{j-1}, list_{leaf_i}[jk + jd], i)$
6. remove the first leaf of $list_{leaf_i}$

Next, we establish the complexity for Algorithm 1. Let n_l be the number of nodes at depth l of the generalized suffix tree for the same input sequences as the factor tree where the box-links are being constructed, and $b_p(k,d) = \min\{n_k^p, n_{pk+(p-1)d}\}$.

Proposition 1. Algorithm 1 takes $O(N^2np)$ time and $O(Nb_p(k,d))$ space.

Proof. Step 1, 2 and 4 require $O(N)$, $O(n)$ and $O(p)$ time, respectively. Step 5 requires $O(N)$ time, which corresponds to the creation or update of $Colors$ array. Hence, Algorithm 1 takes $O(N^2np)$ time. The space complexity is given by the number of box-links, which can be upper bounded by $b_p(k,d)$, times its size, which is N. □

The use of box-links achieves a time and space exponential gain, in the worst case analysis, over approaches in [1]. Time improvement is obtained because the information required to jump from box to box in a structured motif is memorized and accessed very rapidly with box-links. Moreover, it is only required to build a k-factor tree, instead of a full suffix tree, or a $pk + (p-1)d$-factor tree, which leads to important space savings.

References

1. Marsan, L., Sagot, M.F.: Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. J. Comp. Bio. **7** (2000) 345–362
2. Sharan, R., Ovcharenko, I., Ben-Hur, A., Karp, R.M.: Creme: a framework for identifying cis-regulatory modules in human-mouse conserved segments. Bioinformatics **19** (2003) i283–i291
3. Segal, E., Barash, Y., Simon, I., Friedman, N., Koller, D.: A discriminative model for identifying spatial cis-regulatory modules. In: Proc. RECOMB'04. (2004) 141–149
4. Allali, J., Sagot, M.F.: The at most k-deep factor tree. Submitted for publication (2003)