Communications to SIMAI Congress,
ISSN 1827-9015, Vol. 1 (2006)

DOI: 10.1685/CSC06136

Stem-Loop Structure Search

Simone Scalabrin, Alberto Policriti, Michele Morgante
Dipartimento di Matematica ed Informatica,
Dipartimento di Scienze Agrarie ed Ambientali
Università degli Studi di Udine
via delle Scienze 206, 33100 Udine (Italy)
scalabrin@dimi.uniud.it

1 Introduction

The nucleic acid folding problem has been studied since the sixties but only nowadays, that entire genomes are available, fast computational methods are required to scan lots of sequences for high throughput stem-loop structure detection. This structure is shared by many families of genomic elements, such as tRNAs, microRNAs, MITEs, pseudoknots and hairpins present at split ends of different elements such as transposons, helitrons and some 5′-UTRs, just to cite a few.

We introduce helitrons as a case study. Helitrons are recently discovered transposable elements (1) which are likely to be present in all eukaryotes. Basically they are constituted by different motives: an `AT` target site, a 5′ short conserved region beginning with `TC`, possibly a certain number of $5'-3'$ gene fragments and a final hairpin just before a short conserved region ending with `CTRR`.

In the case of helitrons, the search of hairpins should be embedded in the search of more complex elements such as structured motifs (2). In general, this could be very useful for the computational identification of a combination of loose signals. In some cases, altogether they could drastically reduce the noise-to-signal ratio compared to single signals search, for example in (3) the authors estimate that about 10,000 non-shared gene fragments of maize, a remarkable 20% of the entire set of gene segments, have been mobilized by helitrons. Nevertheless up to few years ago no one was able to recognize them, mainly because signals were considered singularly and not as a whole.

We propose an algorithm that avoids the use of the lowest common ancestor, builds the suffix tree only for S (making use of matching statistics), solves efficiently the linear constraint on the loop and that can be embedded in tools like SMaRTFinder (2) to search for different kind of structures in parallel to lead to a more comprehensive and, at the same time, a more flexible identification of unknown genomic elements.

We will focus on the alphabet $\Sigma_{DNA} = \{A, C, G, T\}$. DNA molecules are subject to base-pairing constraints, where the pairs are (A, T) and (C, G). The complement of a DNA word w, denoted by $\overline{w}$, is obtained by replacing each letter with its pairing base. In particular $\overline{A} = T$ and $\overline{C} = G$.

We denote $d_L(S_1, S_2)$ the Levenshtein distance between two strings S_1 and S_2.

A *hairpin* over an alphabet Σ is a string $\alpha\beta\gamma$ where $\gamma = \overline{\alpha}^R$, α and γ are said *stems* and β is said *loop*. It is *k-approximated* if $d_L(\gamma, \overline{\alpha}^R) \leq k$.

An *(s,ℓ)-hairpin* is a hairpin with $|\alpha| \geq s$ (stem constraint) and $0 \leq |\beta| \leq \ell$ (loop constraint). It is *k-approximated* if $d_L(\gamma, \overline{\alpha}^R) \leq k$ and any of the two stems is at least s characters long.

Given two strings S_1 and S_2, *ms(i)* is the length of the longest substring of S_2 starting at position i that matches a substring somewhere in S_1 (we do not know where). These values are called *matching statistics* of S_2 on S_1 and can be computed in linear time using the suffix tree of S_1.

The set of boundary nodes over a suffix tree $\mathcal{T}$ and a depth d, $\mathcal{B}(\mathcal{T}, d)$, is the set of the first explicit nodes at string-depth at least d.

Given a string α and the node v in $\mathcal{B}(\mathcal{T}, |\alpha|)$ whose path-label spells at least α, we define L_v as the list of occurrences of α in S (leaves of $\mathcal{T}$) and $\overline{L}_v^R$ as the list of occurrences of α in $\overline{S}^R$.

Exact search of (min_s, max_ℓ)-hairpins with matching statistics

Build a suffix tree on the input string S and use it to compute the matching statistics of $\overline{S}^R$ on S. These two initial computations allow us to determine forward repeats with one component is S and one in $\overline{S}^R$ and this implies we can also determine inverted repeats (corresponding to stems) with both components in S: if an element of a forward repeat is in $\overline{S}^R$ then it has a correspondent element of an inverted repeat in S. Moreover, using matching statistics, we can constrain on the minimum size of such repeats, and this will guarantee the satisfaction of the stem constraint, see figure 1 for details.

To take care of the loop constraint we exploit its linearity: for each possible stem we sort the corresponding lists of occurrences in S and in $\overline{S}^R$ (L_v and $\overline{L}_v^R$). Subsequently, instead of comparing all the elements of one list with all the elements of the other, we consider a window as large as the maximum loop, max_ℓ, and we shift it linearly on the two lists. In this way we reduce the time needed for solving the loop constraint from quadratic to linear (on the number of repeats computed above).

Finally, hairpins found have to be extended to stems longer than min_s.

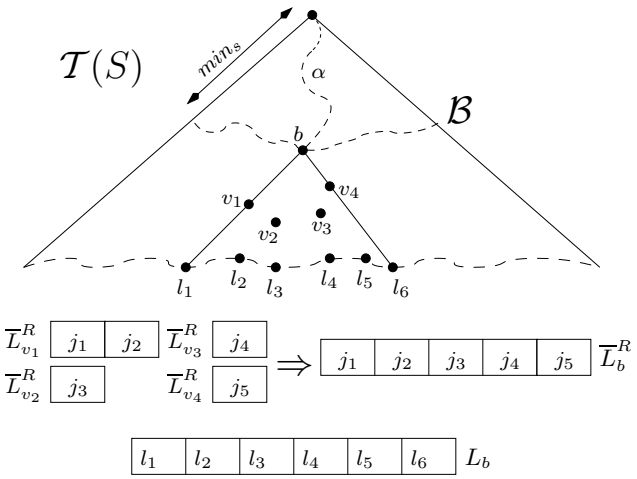

Figure 1: Let the path-label of $b \in \mathcal{B}(\mathcal{T}, min_s)$ be α. Suppose v_1 to v_4 are the only internal nodes rooted at b reached by the matching statistics step: visiting the subtree of b their lists are appended to $\overline{L}^R_b$ (in this case initially empty). During the same visit leaves l_1 to l_6 are inserted into list L_b. At this point all occurrences of string α in S and in $\overline{S}^R$ are listed respectively in L_b and $\overline{L}^R_b$: these are all potential stems labeled by α.

Results and future directions. We implemented a prototype version of our algorithm in C. We tested it on all sequences, at least 20Kbp long, deposited at *nr* and *htgs* databases of *NCBI* of *Zea Mays* to search for the combination of helitron signals.

In the latest column of Table 1 we have the number of sequences belonging to a cluster of at least four elements of the previous column showing high similarity.

Database	Sequences	Hairpin + `CTAGT`	Conserved $3'$ end
nr + htgs	709	30179	1876
nr	68	3022	53

Table 1: Results on Zea Mays sequences at least 20Kbp long (average 150Kbp). Notice that conservation at the $3'$ end does not scale linearly due to the clustering algorithm and parameters.

Results and performances are promising. In particular even though a low threshold has been used to cluster putative helitron (four sequences), larger clusters of up to around 100 sequences are present. On at least one of such clusters, extensive verification has taken place and has given positive outcome. We still have some difficulties in defining the $5'$ region of hair-

pins since it is not well conserved, not even in members of the same family, regardless of the $5'$ signature `ATC`. Anyway, the first results look very promising and the strategy of searching for a combination of different, even though loose, signals seems a winning one. So our first goal is to improve the current implementation and to embed it into SMaRTFinder to reduce the human interaction and speed-up the process. A successive goal is to build a module for SMaRTFinder to recognize gene fragments, often present in helitrons.

In the search of inverted repeats our strategy overcomes space-efficiency problems of other tools. In practice we generate a limited form of a generalized suffix tree for the input string S and for $\overline{S}^R$. The complete suffix tree is built only on S, while for $\overline{S}^R$ we store only the information we are interested in: stems longer than a minimum size. It is only this information which is saved in $\overline{L}_b^R$ along the nodes b of the boundary set. Consequently, our approach enables us to reduce, by a factor of two, the memory requirements in the construction of the suffix tree. Moreover, in the search of approximate repeats, instead of using the highly memory consumptive *lca* labeling on the tree to extend initial seeds (i.e. exact matches), we plan to exploit the already available information of matching statistics to build an exclusion method.

We emphasize that our algorithm has been designed not only to save memory on the construction of the suffix tree but also to handle efficiently in space and time the constraint on hairpins and on similar structures as knots and pseudoknots. Efficiency comes from the fact that the loop constraint is solved locally at each node of the boundary set after sorting its lists of occurrences, without reporting and verifying the presence of the loop for all inverted repeats at the end.

REFERENCES

1. V.V. Kapitonov and J. Jurka, Rolling-circle transposons in eukaryotes, *Proc. Natl Acad. Sci. USA*, vol. 98, 2001, 8714–8719.

2. M. Morgante and A. Policriti and N. Vitacolonna and A. Zuccolo, Structured motifs search, *J. Comput. Biol.*, vol. 12, 2005, 1065–1082.

3. M. Morgante and S. Brunner and G. Pea and K. Fengler and A. Zuccolo and A. Rafalski, Gene duplication and exon shuffling by helitron-like transposons generate intraspecies diversity in maize, *Nature Genetics*, vol. 37, 2005, 997–1002.