ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
CONSIGLIO NAZIONALE DELLE RICERCHE

G. Fiscon, P. Paci, T. Colombo, and G. Iannello

STRUCTURAL ANALYSIS OF LONG NON-CODING RNAs

R. 13-21 2013

Giulia Fiscon - Institute for System Analysis and Computer Science "Antonio Ruberti" (IASI), CNR, Viale Manzoni 30, 00185 Rome, Italy. Email: giulia.fiscon@iasi.cnr.it.

Paola Paci - Institute for System Analysis and Computer Science "Antonio Ruberti" (IASI), CNR, Viale Manzoni 30, 00185 Rome, Italy. Email: paola.paci@iasi.cnr.it.

Teresa Colombo - Institute for Computing Applications "Mauro Picone" (IAC), CNR, Via dei Taurini 19, 00185, Rome Italy. Email: teresa.colombo@gmail.com.

Giulio Iannello - Centro integrato di ricerca, Università Campus Bio-medico di Roma, Via Alvaro del Portillo 21, 00128, Rome, Italy. Email: g.iannello@unicampus.it

ISSN: 1128-3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

The sequence-structural alignment of RNAs is a challenging and computational onerous issue in the field of structures prediction and study of functional RNAs.
To date, the available tools for computing structural alignments are either based on heuristic approaches and thus produce suboptimal alignments or cannot handle instances of reasonable input size. The topic appears more challenging with the incoming of long non-coding RNAs (lncRNAs), a novel set of transcripts whose role in post-transcriptional regulation has been recently shown to be far more prominent than initially believed. Belonging to the huge family of RNAs that does not encode a protein, with a number of nucleotides longer than 200, lncRNAs fold in complex secondary structures and yet little is known about their biological function.
We focus on their secondary structures thanks to which they can recruits different protein complexes. The latter particular functioning can constitute their revealing signatures of functional RNAs to search for. Thus, if two molecules are functionally related and have similar structures, it allows to draw conclusions about the structure of the unknown molecule or about function of molecule not yet functionally characterized.
Our aim is to infer the mechanism of action of uncharacterized lncRNAs, looking for their shared secondary sub-structures with functionally characterized ones. To this end, the structure of a lncRNA with known function can be used as the reference to run pair-wise comparisons with all the others of unknown function. In such an approach, there are two main step involved: first, to assign a structure to the reference lncRNA and second, to look for structural motifs that match the ones characterizing the reference lncRNA.
We developed two novel algorithms able to handle the two following tasks: structure prediction and matches evaluation, in order to improve or replace widespread tools, which are not suited to exhaustively deal with lncRNAs.

1. Introduction

The last years have been the scene of an increasing interest on long non-coding RNAs (lncRNAs), which are antisense, intronic and intergenic transcripts longer than 200 nucleotides [1, 2, 3, 4, 5, 6]. Key elements for understanding human evolution, development and cognition, critics both in the transcription and post-transcriptional mechanisms (as splicing, editing, transport, translation and degradation of their mRNA transcripts) and guide in the processes of epigenetic regulation and chromatin modifications, the lncRNAs are a very interesting surprise, that recently became more and more challenging [7, 8, 9, 10].

There are tens of millions of ncRNAs of which many have been described and identified by chance in the years 90 and the evidence that they are found abundantly transcribed somewhere in the genome and that are transcribed in a constant, regulated in time and space strengthen the idea of their actual involvement in developmental biology, human disease and regulatory processes in eukaryotic cells [11]. The *lncRNAs* are mostly defined as "dark matter"', because only a part of them is associated with a specific function. They could only be considered "transcriptional noise" due to their low level of species conservation in the genome. However, a possible explanation for this lack of conservation may be the rapid evolution of the sequences of lncRNA, in addition to the finding that these molecules do not require a high rate of conservation of the nucleotide sequence, in order to maintain their functionality. In fact, the genes encoding proteins are object to a specific selection since they need to maintain the correct coding of amino acids and open reading regions; whereas the RNA molecules have less strict sequence requirements to keep their normal function and their flexibility means that they are mostly candidate to evolutionary changes of genes coding. Furthermore, if the transcription of lncRNAs was only "noise", the levels of expression of the transcripts lncRNA should not vary spatially, temporally or in response to stimuli as instead happens. Thus, if it has been shown that evolutionary conservation is a symptom of functionality, its absence does not imply the absence thereof in such non-coding RNAs [12].

These RNAs are single-stranded, however, due to the length of the primary structure (sequence) and RNA instability, they tend to fold back on themselves and originate complex secondary structures. Since little is known about their biological function yet, an effective way to try to predict whether lncRNAs have a role in the cell is to use the computational models that try to determine which sequences of lncRNA form secondary structures such as repeated short stem-loops (couple of paired and unpaired bases, respectively). We are interested on their secondary structure that seems to be conserved along the organism evolution, instead of their primary one. Moreover, the formation of the secondary structure is critical because through these structures that lncRNAs interact with DNA and proteins.

More specifically, along the huge world of lncRNAs we focus on a small part of these, called *lincRNAs* (long intergenic non-coding RNAs), with particular emphasis on finding recurrences of structural motifs, related to their functional role. There are 2458 lincRNAs present in the mammals genome, identified through the all long non-coding transcripts contain in the database known as *RefSeq* (NCBI "Reference Sequences"), *Ensembl* and *UCSC Genome Browser* [12, 13].

Becoming aware of the great importance of the lncRNAs structure, we developed a method of analysis and evaluation for comparing two lincRNA secondary structures. Our aim was to functionally characterize lincRNAs of known sequence and structure, but of unknown function,

by comparing their primary and secondary structure with the one of RNAs with known functions. The driving idea was that similarity in secondary structures might imply common functions.
While similarity between two nucleic acid chains is usually determined by sequence alignment algorithms, but since these can only account for the primary structure and thus ignore structural aspect, we approached these issue in another way: not performing a structural alignment ([14] and references there-in) but looking for a potential structural folding within a candidate lincRNA sequence.

Starting from the sequence of *a reference lincRNA* (the lincRNA whose function is known and linked to the structure), our structural analysis method (i) predicts the reference secondary structure (made up of different structural elements) and maps it in the following form using a dot-bracket notation; (ii) extracts the single RSSPs (RNA Sequence-Structure Pattern) making up the entire reference structure, aiming to look for them in the sequence of the *target lincRNA* (the lincRNA whose function has been unknown and which could share same function based on same common sub-structures); and finally (iii) evaluates the obtained groups of non-branching structures.

Among the relevant existing tools, *RNAfold* [15] is currently the most popular tool used to predict RNA secondary structures based on free energy minimization, while *Structator* [16] is the leading one to search RNA secondary common structural motifs. However, the present task unveiled some limitations concerning applicability of these tools that we had identified and addressed. Firstly, the lack of communication between these ones: *RNAfold* provides the entire structure made by hairpin, internal loops, bulges, multi-loops while *Structator* is able to analyze only non-branching structures. Secondly, Structator uses only the number of patterns found as a search criterion, not providing enough information about the quality of the alignments identified.

Here we present two novel algorithms developed to overcome these limits: the first one able to extract, given a sequence of one lncRNA and its structure predicted by *RNAfold*, the non-branching sub-structures that can be used as input for *Structator*; the second one able to define a score for the alignment between the RNA Structure-Sequence Patterns (RSSPs) in the reference and target lncRNA.

2. Material and Methods

2.1 Algorithm to extract RSSP (RNA Sequence Structure Pattern)

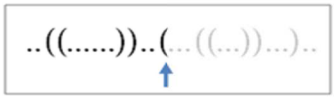
We developed an algorithm that provides, given a sequence of a lncRNA with the corresponding predicted structure in the Vienna format (dot bracket notation), the non-branching sub-structures contained therein. These sub-structures are then saved in an appropriate file according to the syntax required by Structator tool.

2.1.1 Extraction of the non-branching structures

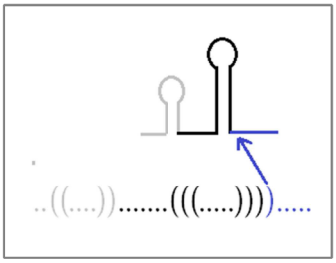
The algorithm needs to scan a structure in dot-bracket notation and correctly identifies the non-branching structures. It treats of realizing a simple parsing of the string since the non-branching structures are not referable to regular expressions: indeed it has to be able to recognize structure like the following one: (((..((.....))...))), where the number of closed bracket has to be the same,

instead of the number of couples that can be anyone. The leading idea is to recognize the potential start of a non-branching structure and to store the position of an open bracket until it finds the closed one.
The presence of a closed bracket after one or more open corresponds to an external loop, or rather to a non-branching structure. Once an external loop is identified, continuing the scan, the non- branching structure can halt:

- because it met an open bracket, pointing the potential beginning of a new non-branching structure;



- because all open bracket encountered by the potential start of the scanning non-branching structure have found their corresponding closed ones; in this case, in fact, any other closed brackets correspond to a part of secondary structure from which more non-branching structures begin.



The algorithm has one for-loop and it can be found in three different states, called open, closed and undecided, as showed in the Figure 1.

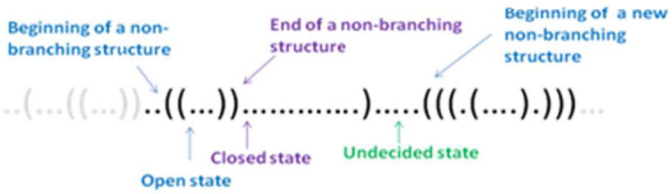
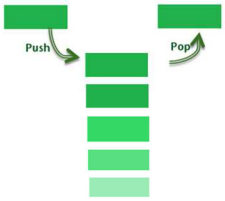


Figure 1 Graphic representation of the algorithm: During the scansion it can be meet three different states: **open state**, in which it begins to scan the string and it's finding only open brackets (i.e., the beginning of a non-branching structure closed); **closed state**, in which, continuing the scansion, it begins to see the brackets, that are paired whit the previously open ones allowing to define a non-branching structure; **undecided state:** along the structure it has found the end of a structure not branching, but not yet the beginning of the next one. In this case, as soon as it meets with an open bracket, it starts the parsing of a new non-branching structure and then goes to the open state, whereas as long as it meets closed, it has to skip them and go on.

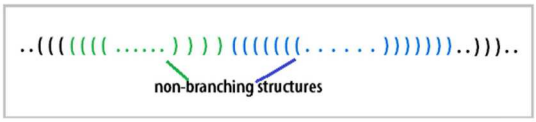
We used a start position and a stack as data structures: the "start position" is a variable that stores the beginning point to looking for a new structure, although not necessarily coincides with the start point of the non-branching structure; the stack implements a LIFO policy (last-in, first-out), where the element deleted from the set is the one most recently inserted. Specifically in our case the "push" operation takes place when it encounters an open bracket, storing the index of the current position, while the "pop" operation acts out when it meets the

corresponding closed ones, until the stack is not empty, indicating the end of the non-branching structure.



Now we focus on some particular cases that had to be addressed, whose solution is shown.

- Firstly, in such case when it has been encounter an open bracket, lying in the closed state, the algorithm has to check whether the stack is empty or not, because can occur that more non-branching structures are nested within of a larger structure, which therefore should not be included in the computation of the resulting structures. It is to insert an additional control when it has come to find a possible structure, checking that for each closed parenthesis found there is its corresponding open one, determining the proper base pairing.




```
% FOUND STRUCTURE
elseif state = CLOSED
    if R(i)=='('
        if stack.empty
            pairs.append((start,i-1))%the beginning of the
            structure is "start"
            start=i;
            state = OPEN;
            stack.reset;
            stack.push(i);
        else
            pairs.append((stack(end)+1,i-1)) %the beginning of
            the structure is the stack head
            start=i;
            state = OPEN;
            stack.reset;
             stack.push(i);
        end
...
```

- Secondly, the case of encountering a closed bracket being always in the Closed state, we need to check if the stack is already empty or not, because only in the first hypothesis it will be found a non-branching structure.

```
elseif state = CLOSED
        ...
       elseif R(i)== ')'
             if isempty(stack)
             % FOUND STRUCTURE
                 pairs.append((start,i-1))
                start=i+1;
                state= UNDECIDED;
             else % if the stack is not empty
                stack=stack(1:(end-1)); % extract stack head
             end
       end
```

- Finally, it is necessary to manage the issue of whether the last structure is included when the scan ends (otherwise would be lost). Thus, it has been checked the state of the algorithm when the end of the entire structure to be scanned is reached: if it is “closed” it will mean that we have one more non-branching structure to add.


```
%% issue of the last structure

if state=CLOSED % other structure has still been added
pairs.append((start,n))
end
```

Examining the whole algorithm appears clear how it scans the string elements at least once. Therefore his computational complexity is linear with the length of the sequence to be analyzed.

```
% state OPEN/CLOSED/UNDECIDED
% pairs: list of index couples

stack.init;
pairs.init;
R = read the structure string of the file;
start = 1;
state = OPEN;
n= lenght(R)
for i=1:n
        if state = OPEN
                if R[i]= '(' % meeting an open bracket
                        stack.push(i);
                else R[i]= ')' % meeting a closed bracket
                        stack.pop ;
                        state = CLOSED;
                end
        else if state = CLOSED
                if R[i]= '('
                        %found_structure
                        if stack.empty % stack is empty
                                pairs.append((start,i-1))
                                start=i;
                                state = OPEN;
                                stack.reset;
                                stack.push(i);
                        else % stack is not already empty
                                pairs.append((stack(end)+1,i-1)) % the beginning of the structure is the following position of the
                                last open bracket (stack head)
                                start=i;
                                state = OPEN;
                                stack.reset;
                                stack.push(i);
                        end
                else R[i]= ')'
                        if stack.empty % stack is already empty
                                %found_structure
                                pairs.append((start,i-1))
                                start=i+1;
                                state = UNDECIDED;
                        else % stack is not empty
                                stack.pop;
                        end
                end
        else if state = UNDECIDED
                if R[i] = '(' % beginning of a new search
                        stack.push(i);
                        state = OPEN;
                else R[i]=')'
                        start=i+1; % state remains UNDECIDED and skip it
                end
        end
end
```

Figure 2 Pseudo code to perform the part of the algorithm which recognizes and extracts the non-branching structures.

2.1.2 Generation of the Output Files

Once the non-branching structures have been properly identified and stored in the appropriate data structures, it was necessary to deal with the generation of the final output file, which could be accurately recorded. We aim to get the file that specifies all RSSPs within the entire structure

of the analyzed lncRNA, which made up the Secondary Structure Descriptor (SSD), required as an input of Structator. Since we have to capture the sequence-structure similarities but we do not have the evidence about structures with special constraints, the idea is to look at first only the pure structure, in order to find all possible occurrences. Then, we will want to investigate more deeply the common elements and it is therefore necessary to provide the possibility to insert some constraints that make our searching more specific, such as whether pairing of particular nucleotides.

It has been necessary to generate two files, consistent with the syntax required by the program Structator, which differ in the row for the sequence (i.e., unspecified in one of these files), as showed in Figure 3 and Figure 4. The first three lines of output files are so defined: the first one contains the name of the pattern found and the search parameters assigned; the second one contains a sequence of characters called "wild card" in that they can match more than one nucleotide; the third one presents the non-branching structure in the Vienna format.

Figure 3 File output 1

Whereas the first and the third row of the second file remain the same, the second row shows the sequence corresponding to the original not-branching structure instead of the "wildcard" characters.

Figure 4 File output 2

The two output file (Figure 3 and 4) are computed trough three steps (Figure 5,6, and 7).

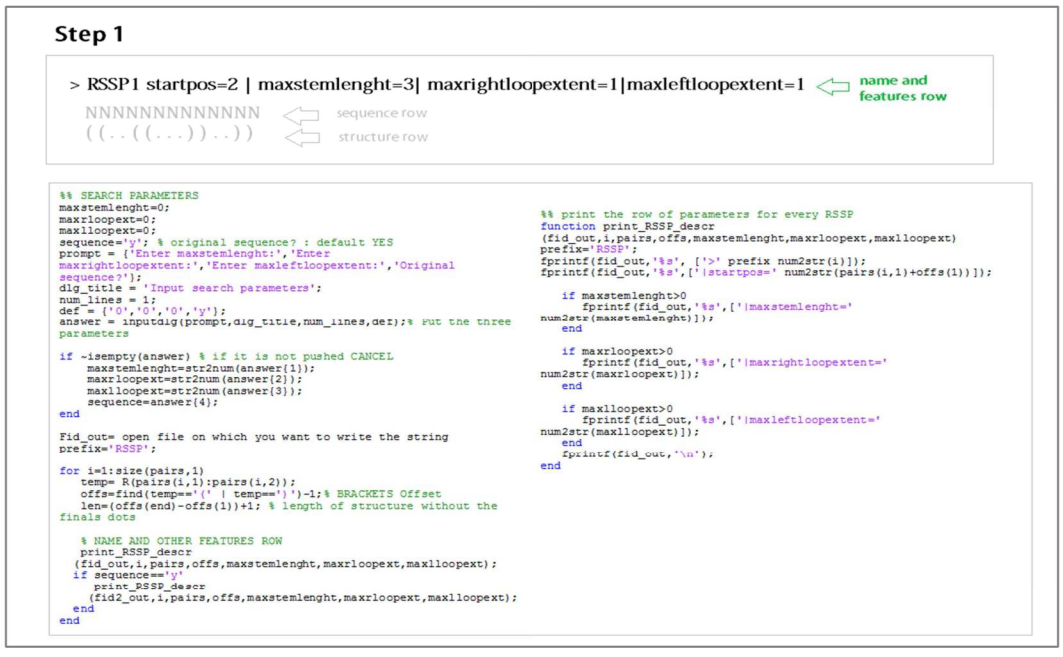


Figure 5 Step 1: The row of RSSP name and features. The line starts with the *pattern name* of the non-branching structure is composed of a common prefix (RSSP) followed by a distinctive number linked to the order with which has been found within the input structure. To follow, the search parameters demand by Structator are reported **[16]**: *startpos* (i.e., the starting position of the non-branching structure in the original sequence) *maxleftloopextent, maxrightloopextent, maxstemlenght* (i.e., the maximum length of the loop from the right and from the left and the maximum length of the stem, respectively) which confer a flexibility in the structures reproduced, adding up to a specified number of wildcard characters to the right and left of the loop or at the end the stem and simultaneously in both.

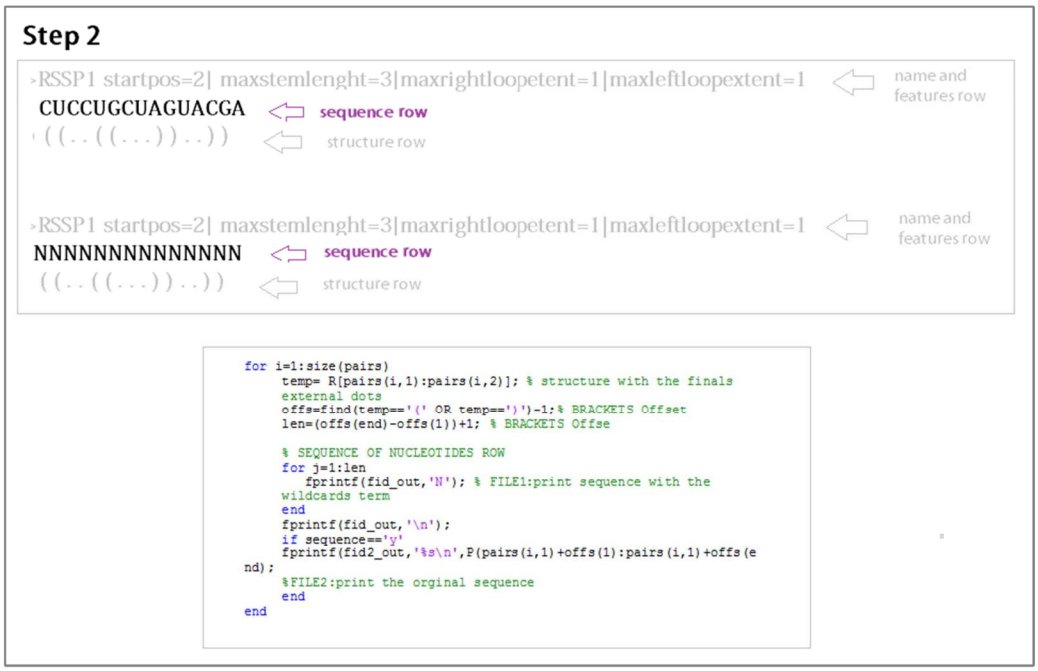


Figure 6 Step2: the sequence row. It may present, unless the user specifies otherwise, both the sequence of nucleotides originating, both a sequence of wildcard characters, corresponding to any nucleotide.

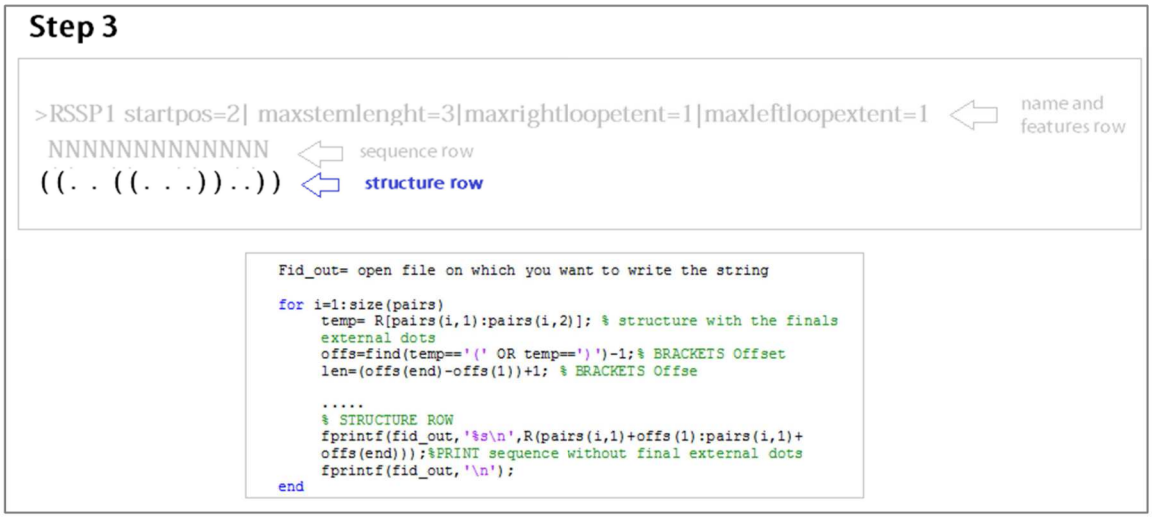


Figure 7 Step 3: Structure row. They are printed on both the output files the final non-branching structures of any RSSP in dot-bracket notation (it should be noted that they are printed without the final external dots, called " queue").

2.2 Algorithm to evaluate and label Secondary Structure Descriptor (SSD)

Structator provides outgoing occurrences found (matches) and one or more alignments of RSSP (chains) of lincRNA analyzed. A chain is therefore a sequence of matching of RSSP found in the same position of the reference sequence. However, since a limit of Structator was to use as a search criterion only the number of the patterns found, not providing enough information about the quality of the alignments identified, it has been needed develop a *score algorithm*.
In order to investigate the most significant subparts of a chain, the evaluation function that has been developed is based on the "gap", i.e., the distances between the individual RSSP aligned in lncRNA reference, compared with those in which these ones are in lncRNA target. The metric that has been used regards the average of the deviation between the gap of contiguous subparts, considering, however, not only the close pairs, but all possible contiguous groups.
Indeed, if we considered only the distance between the RSSP taken two by two and close, there would not be reliable quantitative analysis of the result. For example, the difference between two alignments would not be highlighted properly, e.g., in both there could be not negligible distances between certain groups of contiguous RSSPs, and only one may be present RSSP properly aligned, which would not weighted. On average, the two arrays would be similar, but would not show those part of correct alignment which makes one better than the other.

First of all, starting from the Structator output we built up the *cost matrix* of all possible pairs: the (i,j) element of the matrix corresponds to cost pair (i,j), as showed in Figure 8.

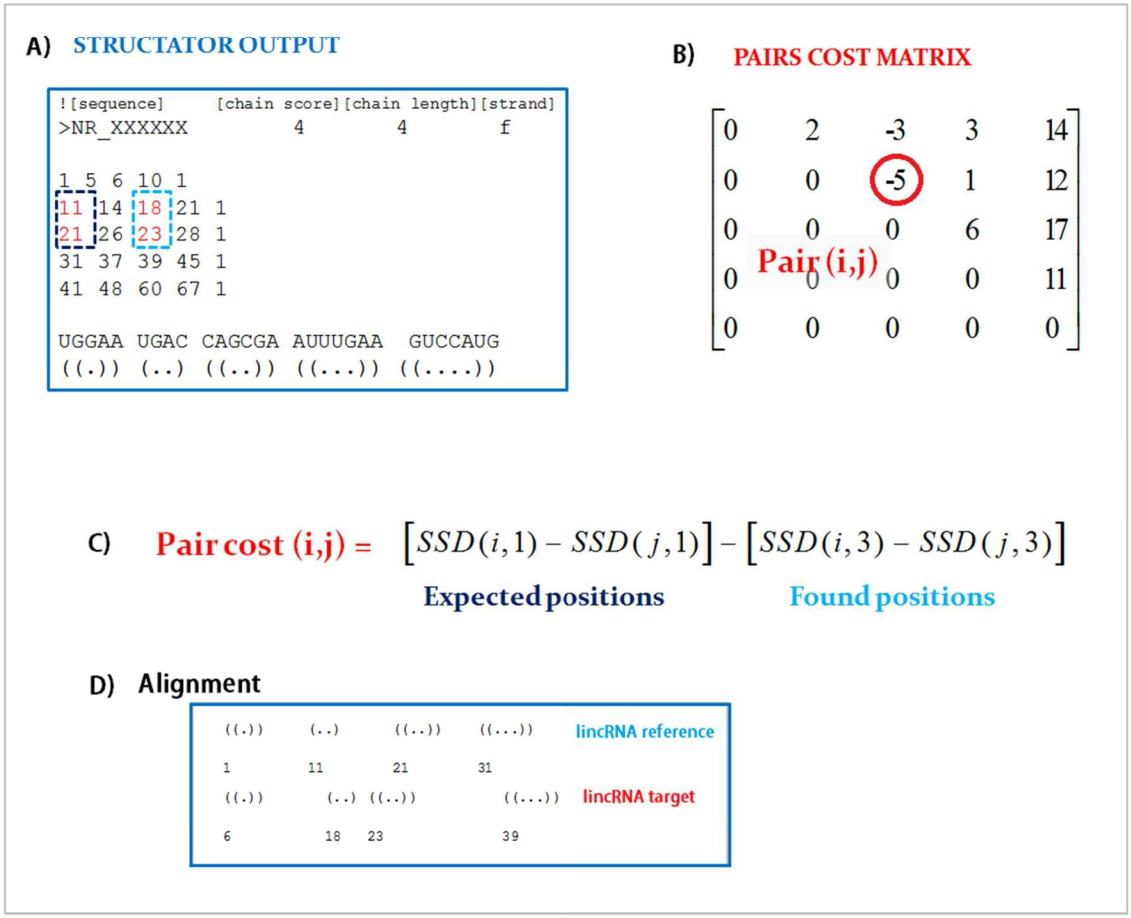


Figure 8 Building up the Cost matrix. A)Output of Structator: the first line presents the sequence description with some auxiliary information (name, the length and the score assigned to the chain, the direction of the scan, i.e., f = towards direct); to follow the coordinates of RSSP, i.e., the start and end of the expected positions, corresponding to reference lincRNA and those found in target lincRNA; finally the last numerical value corresponds to the weight

associated with each fragment of the chain (=1 in this example). The end of the file has the substring consisting of the RSSP which occurred the match. **B)Cost matrix:** starting from the file (A) we build up a matrix with the cost of all possible pairs, using the formula showed in (C). **C)Formula to compute the pair cost (i,j)**: subtraction between the expected and the found positions, obtained by the file (A) and necessary to build (B) up. **D)Example of the alignment** corresponding to the values presented in (A): we show graphically the chain of the reference RSSP and the target chain with position of the same RSSP found in the target lincRNA (it has been canceled the relative distance between the initial positions of the reference and target lincRNA).

To follow, from the cost matrix we extracted firstly the over-diagonals elements, which will constitute the weight of the gap of close pairs. Then, taking account of all contiguous subsets who made up the analyzed SSD, we constructed a further matrix (*score_matrix*) where are showed the scores assigned, which correspond to the average error calculated from the cost matrix (Figure 9).

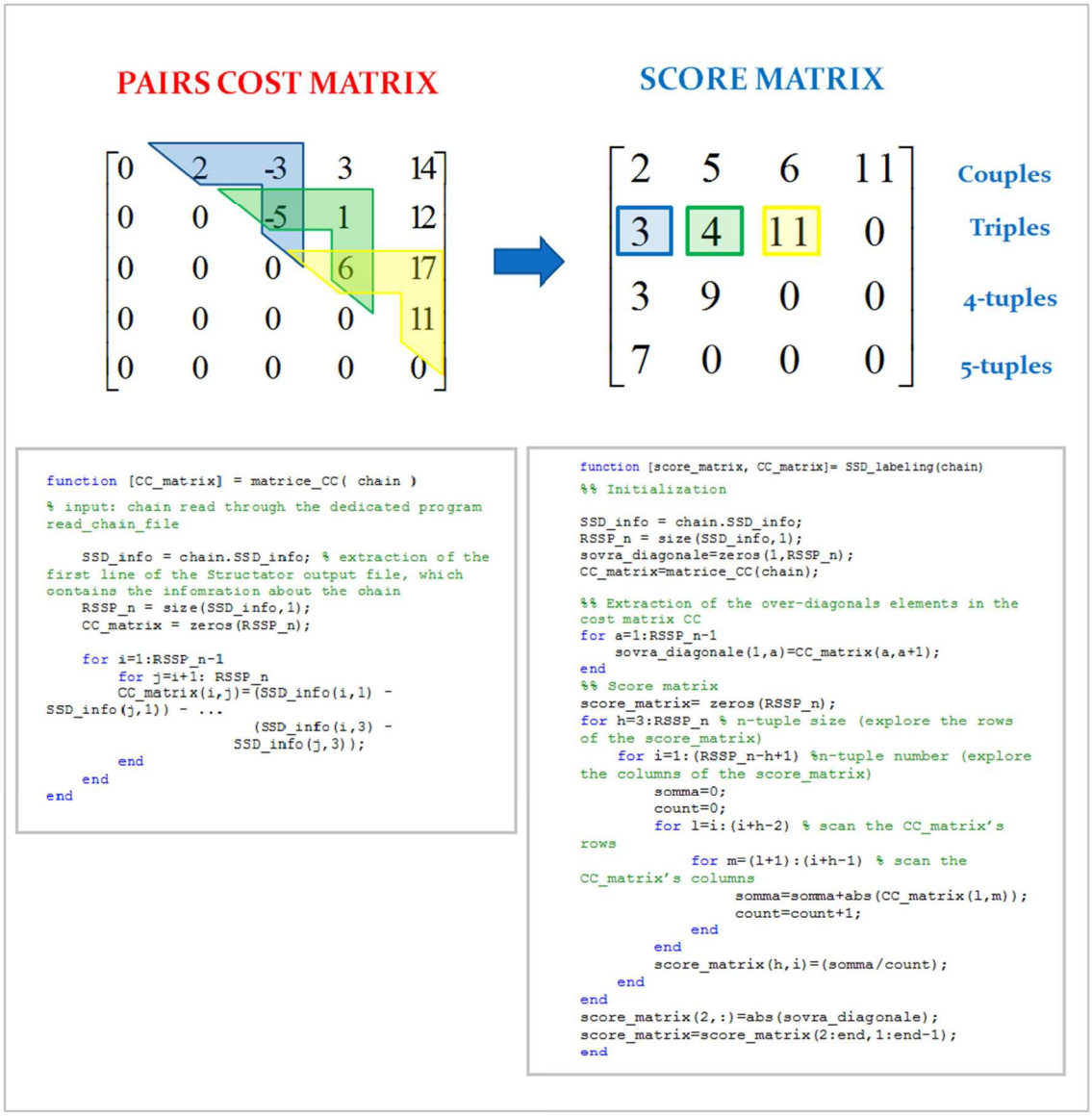


Figure 9 Building up the Score-Matrix: Built the cost matrix, it has been constructed the *score_matrix* with the scores assigned (i.e., the average error calculated from the cost matrix): starting from the pairs up to the *n-tuples*, it has been covered the entire SSD. Let *n* the number of all element and *h* the size of the *n-tuples* , then the number of *n-tuples* is *h*+*n*-1. In this picture for example we showed the construction of the triples score, highlighted in the color blue green and yellow.

The complexity of this algorithm is polynomial compared with the length of the analyzed SSD (i.e., the number of its component RSSP). In particular, it can be shown that it has complexity $O(n^4)$ with a multiplicative constant much less than 1

.

3. Results and Discussion

Among the lincRNAs whose function is related to their structure, we selected *Hotair* [17] (intergenic antisense transcript, located in the cell nucleus), *Coldair* [18] (intronic transcript result of a significant biological event), *Anril* [18] (antisense transcript) and *Kcnq1ot1* [17], where most likely the function is due to the structure and in particular the structure might have a common theme, since they all interact with the same protein complex [19], [20]. In fact, they assume the role of "scaffold" or "guide" for the chromatin-modifying protein complexes linked to epigenetic regulation. In this regard, it was recently shown that Anril and Coldair act in very similar way to Hotair interaction with the protein complex, called PCR2: Anril in the role of "anchor" (scaffold) to the silencing of target genes, and Coldair in the role of guide, determining the epigenetic repression of genes as FLC (floral locus C) [18]. Moreover, even the *Kcnq1ot1* transcripts recruits repressive chromatin modeling complex such as PCR1 and PCR2 through which epigenetically silences the Kcnq1 imprinting control region.

In order to follow out a correct structural analysis, we developed two procedures (Figure 10), which respond to complementary purposes. The first procedure allows to search consensus structures of a reference lincRNA within the target lincRNA, whose only the sequence is known. Whereas the second one provides an independent tool to assess how significant the RSSP present in the alignments found. The driving idea it is to check if the alignments considered significant based on the results of the first procedure are equally feasible from an energy point of view.
This proposed method of analysis and evaluation is able to highlight subparts of the sequence that have structural characteristics similar to those used to interrogate them.

We applied the two procedures to different combination of the reference lincRNA and target and we show the results obtained.
We considered the follow-up lincRNAs, whose their structure is linked to the function performed [17]: *Hotair*, *Anril*, *Coldair*, *Kcnq1ot1*, and we use one by one the first three as reference. We show the full performance regarding *Hotair* (taken from hg18, Genome Browser) as the reference lincRNA and the left ones as target (Figure 11).

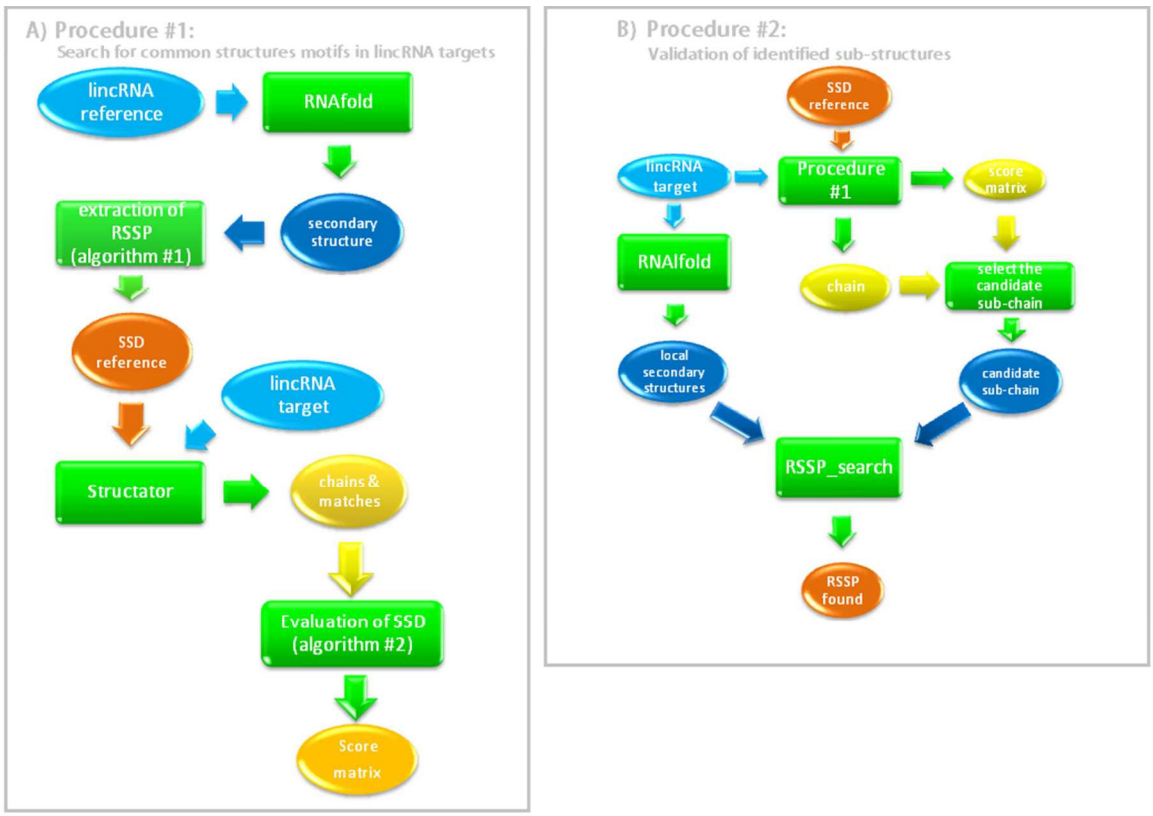


Figure 10 Flowcharts of the two procedures. A) Procedure #1: a structural analysis to search for common structures between the reference and target lincRNA. Once the sequence of reference lincRNA is known, the secondary structure is predicted, using *RNAfold* **[15]**; such prediction is given as input to our algorithm for the extraction of non-branching structures in order to obtain the SSD. Then, the SSD of the reference lincRNA and the sequence of the target lincRNA are given as input to *Structator* **[16]**. The software provides matches and chains that will be evaluated by our algorithm developed to the evaluation of the chains. Therefore, given the SSD reference and the sequence of the target lincRNA as input, the procedure #1 outputs the found alignments (chains) and the score assigned to them. **B) Procedure #2:** looks for the RSSP engendered from the procedure #1 and carefully selected (select candidate sub-chain) in those obtained from the program *RNALfold* **[15]**. We identify two parallel branches: (i) the first branch concerns the procedure #1 application and it focuses on the most significant subparts of the chains (sub-chain) returned from procedure #1, or rather it is selected the best candidate among the latter (i.e., the sub-chain with a greater number of RSSP and with a lower average error); (ii) the second branch is characterized from the application of *RNALfold*. The latter will generate the local secondary structures of the target. Through the comparison between the outputs obtained by the two branch (the local secondary structures and the sub-chain candidate), we select the found RSSP, which will be those really stable based on free energy, index of a plausible biological significance. Legend: The rectangles represent the software used, the circles the outputs of each of them, the ovals the files FASTA of the reference lincRNA and target. Our algorithms are *extraction of RSSP* and *evaluation of SSD*.

3.1 Procedure #1: Search for common structure motifs in target lincRNAs

1) Folding

The sequence of Hotair (reference lincRNA) is given as an input to RNAfold, which provides the prediction of his minimum free energy secondary structure.

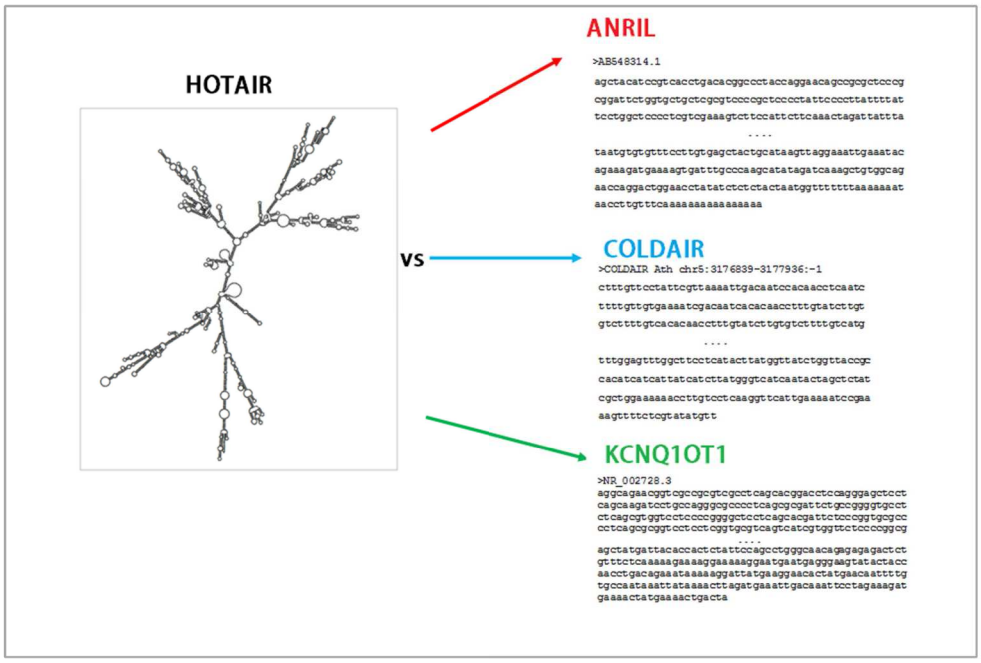


Figure 11 Prediction of Hotair structure VS sequence of Anril, Coldair, Kcnq1ot1.

2) Extraction of RSSP

Trough our first algorithm we extracted the non-branching structures, which constitute the pattern to be provided in input to Structator together with the sequence of the target. We identified 45 RSSP (three shown in the Figure12) which made up the Hotair secondary structure and which were then searched for in the other three lincRNA, target of the test: Anril, Coldair and Kcnq1ot1. Note that we considered the sequence with the nucleotides not specified ("wildcard" N), since in this step we were interested in the search of the "pure structure" inside the target.

```
>RSSP1|startpos=10
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
(((((.(((((((...)))))))..)))))

>RSSP2|startpos=71
NNNNNNNNNNNNNNNNNNNNNNNNNNNN
(((((((((.......)))))).))))

-----------

>RSSP45|startpos=2263
NNNNNNNNNNNNNNNNNNNN
(((((((.....)))))))
```

Figure 12 Hotair SSD: set of the 45 RSSP extracted by algorithm#1

3) Looking for common structural motifs in other target lincRNA by Structator

We searched for the set of 45 Hotair RSSP (Hotair SSD reference) in the three target lincRNA. Based on results produced by Structator (matches and chains in Figure 13) we note that in most cases there are also occurrences for more than half of the pattern sequence-structure.

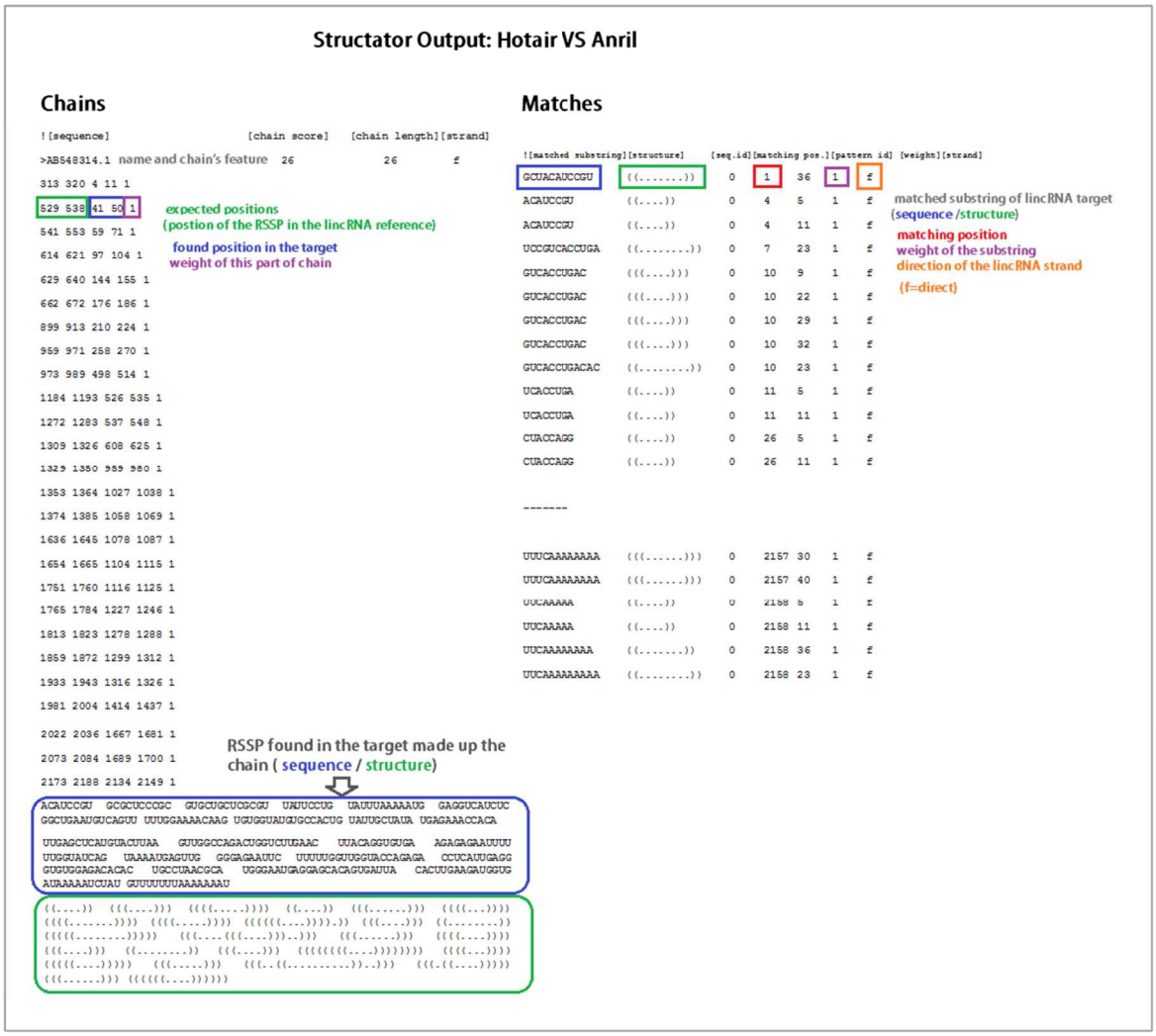


Figure 13 Example of Structator output: chains and matches found looking for SSD reference (Hotair) in the target (Anril): 1.Chains: it can be distinguished three part (i) the first line reports the name and the features of the identified chain (score, length, strand direction), (ii) the second line show the coordinates of the RSSP (i.e., the start and end of the expected positions, corresponding to Hotair and those found in Anril) and the weight associated with each fragment of the chain (=1 in this case); (iii) finally the end of the file presents the substring consisting of the RSSP found in Anril that made up which the chain. **2.Matches:** it reports the matching sub-string of Anril (sequence follow by structure), the matching position, the weight of the sub-string and the direction of the Anril strand.

4) Evaluation of reported matches

In order to obtain a reliable measure of similarity, once found the alignments, it has been applied to the latter our second algorithm, or rather the Score Matrix for all matches. The analysis of the matrices produced by considering the triples up to the total tuples, it was found that there were of the subparts (sub-chain) with best matches. The significance of the subset has been obtained taking into account both the length, i.e., the number of RSSP aligned, both the average error, i.e., the score. There is therefore the best sub-chain with greater length and with a

score lower, but highlight the longest ones, even if with worst score. In the specific case we focused on sub-chain with a score of less than or equal to 50 and when Structator used to provide more alignments for the same target, since the difference observed between these was small, we considered only the main matches. In Table 1 we report the results thus obtained. It was applied the same procedure but regarding as reference lincRNA Coldair, from which we were extracted 23 RSSP and Anril, from which we were extracted 38 RSSP (the value of score also in this case is equal to or less than 50). The results are shown in Table 2 and 3.

Table 1 Search results of the procedure#1 for RSSP of reference lincRNA Hotair: the best sub-chain illustrate the pairs of values (L, S) found, where L is the length of RSSP and S is the score.

LincRNA reference VS target	N°Matches/N°RSSP reference	N° chains	Chains length	Best sub-chain (L,S)
Hotair VS Anril	26/45	2	26	(8,41) (5,38) (5,44) (3,18) (3,23) (3,21) (3,36) (3,38)
Hotair VS Coldair	25/45	1	25	(5,27) (3,14) (3,17) (3,29) (3,31) (3,37) (3,43)
Hotair VS Kcnq1ot1	35/45	19	35	(3,34) (3,36) (3,39) (3,44)

Table 2 Search results of the procedure#1 for RSSP of reference lincRNA Anril: the best sub-chain illustrate the pairs of values (L, S) found, where L is the length of RSSP and S is the score.

LincRNA reference VS target	N°Matches/N°RSSP reference	N° chains	Chains lenght	Best sub-chain (L,S)
Coldair VS Anril	12/23	7	12	(3,32) (3,37) (3,46) (4,41)
Coldair VS Hotair	12/23	5	12	(4,44) (3,22) (3,37)
Coldair VS Kcnq1ot1	21/23	37	21	(3,30)

Table 3 Search results of the procedure#1 for RSSP of reference lincRNA Coldair: the best sub-chain illustrate the pairs of values (L, S) found, where L is the length of RSSP and S is the score.

LincRNA reference VS target	N°Matches/N°RSSP reference	N° chains	Chains lenght	Best sub-chain (L,S)
Anril VS Hotair	22/38	17	22	(4,46) (3,20) (3,21) (3,42)
Anril Vs Coldair	12/38	13	20	(5,50) (4,14) (3,10) (3,14) (3,16) (3,19)
Anril VS Kcnq1ot1	31/38	7	31	(4,45) (3,16) (3,22) (3,25) (3,45)

3.2 Procedure #2: Validation of identified sub-structures

The second procedure (see Figure 10) has been applied to all the previous cases, and particularly in those subsets that are most significant results, on the basis of what has been obtained from the first structural examination (last column of Tables 1, 2 and 3). It gives a detailed analysis of the sub-chain obtained by considering the pair Hotair-Anril (as reference lincRNA and target respectively), since it is particularly interesting for its length.
Looking for the RSSP of the sub-chain candidate among those RNALfold, we have found five that perfectly match, which are shown in Figure 14. Having found more than one match with local structures above could therefore confirm the reliability of the first procedure, recognizing the best RSSP considered as plausible candidates from the point of energy.

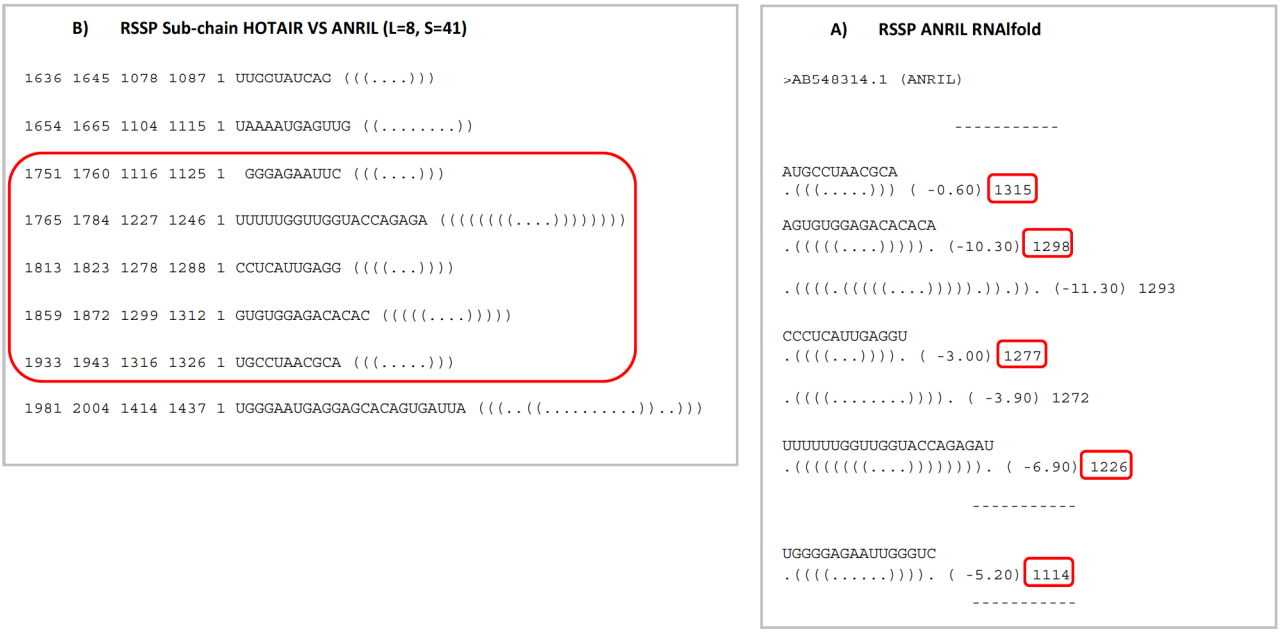


Figure 14 Search of the RSSP of the target lincRNA (Anril), aligned with the reference lincRNA (Hotair), among those predicted by RNALfold. (A) The Anril sub-chains obtained by the alignment with Hotair (L = Number of RSSP, S = Mean error): It has been showed the eight RSSP consecutively aligned, identified by the coordinates of the start and end position, respectively expected (or rather whose is in Hotair) and real (whose is found in Anril); these coordinates are followed by the weight associated with each RSSP, by the order and the structure. It has been highlighted those five pattern predicted also by RNALfold in the red boxes. **(B) RSSP predicted through RNALfold, receiving as input the sequence of lincRNA Anril**: it has been presented the predicted local structures, followed by free energy associated to each one and the position at which it was found, highlighting suitably those five coincident RSSP. Note that in the latter figure, instead of the RNALfold output we have been added to the sequences corresponding to the structures found in target lincRNA, to allow a better comparison**.**

4. Conclusion

In this work we approached a very challenging problem of the research in genetics and molecular biology today still largely unexplored: clustering a subclass of non-coding RNAs, which have more than two hundred nucleotides (lncRNAs), based on their common structural motifs and compared them to that small portion that is already functionally characterized. It is a really boundary issue coming out in recent years through the pervasive transcription of the genome, but in respect of which there are still no tools, algorithms models and universally defined, suitable to address it. Thus, this project aims to begin to consider how to achieve such

an ambitious goal. It has been critically examined the state-of-the-art tools able to perform a structural analysis. The latter, starting from the prediction of secondary structure, which has been obtained through the suitable folding algorithms, allow to look for particular structures within RNA functional sequences. We observed, however, some limitations on the ability to quantitatively characterize the result of research.

To this end, we developed some novel algorithms, then implemented in MATLAB, to fill up the lack of communication among the tools and to allow a more extensive results provided by these ones. Afterword the functionality of these algorithms were assessed, applying the whole chain of search to a selected number of lincRNAs and discussing the quantitative results. The proposed methodology of analysis and evaluation is able to highlight subparts of the sequence that have structural characteristics similar to those used to interrogate them: one of its more widespread use would select structural elements on which to perform further investigation and biologically significant.

Aware of the complexity because of the extreme importance of this new frontier of research, this work is to take the first step toward a possible structural and functional characterization of the long non-coding RNAs, providing the criteria and procedures that can be then applied in a systematic way. Thus, placing itself in pioneering perspective, this work is not intended as a final phase, but the beginning of a deep survey that will suitably carried out in the future in order to expand it and edit it.

References

[1] J.S.Mattick, «The central role of RNA in human development and cognition,» *Febs Letters,* n. 585, pp. 1600-1616, 2011.

[2] J.S.Mattick and V.Makunin, «Non-coding RNA,» *Human Molecular Genetics,* vol. 15, pp. 17-29, 2006.

[3] J.S.Mattick, R.J.Taft, K.Pang, T.R.Mercer and M.Dinger, «Non-coding RNAs: regulators of disease,» *Journal of Pathology,* pp. 126-139, 2009.

[4] M.Esteller, «Non-coding RNAs in human disease,» *Genetics,* vol. 12, pp. 861-874, 2011.

[5] P.Sumazin, X.Yang, H.Chiu, W.Chung, P.Guarnieri, J.Silva e A.Califano, «An Extensive MicroRNA-Mediated Network of RNA-RNA Interactions Regulates Established Oncogenic Pathways in Glioblastoma,» *Cell,* n. 147, p. 370–381, 2011.

[6] the ENCODE pilot project, «Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project,» *Nature,* vol. 447, pp. 799-816, 2007.

[7] Tano K and Akimitsu N, «Long non-coding RNAs in cancer progression,» *frontiers in genetics,* vol. 3, n. 219, 2012.

[8] M.B.Clark and J.S.Mattick, «Long noncoding RNAs in cell biology,» *Seminars in Cell &*

Developmental Biology, n. 22, pp. 366-376, 2011.

[9] M.Cesana, D.Cacchiarelli, I.Legnini, T.Santini, O.Sthandier, M.Chinappi, A.Tramontano and I.Bozzoni, «A Long Noncoding RNA Controls Muscle Differentiation by Functioning as a Competing Endogenous RNA,» *Cell,* n. 147, pp. 358-369, 2011.

[10] Qureshu IA, Mattick JS, and Mehler MF «Long non-coding RNAs in nervous system function and disease,» *elsevier,* pp. 20-35, 2010.

[11] P.Amaral and S.Mattick, «Noncoding RNA in development,» *Mamm Genome,* pp. 454-492, 2008.

[12] M.Guttman, I.Amit, M.Garber, C.French, M. F.Lin, D.Feldser and M.Huarte, «Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals,» *Nature,* vol. 458, pp. 223-227, 2009.

[13] I.Ulitsky, A. Shkumatava, C.H.Jan, H. Sive and D.P.Bartel, «Conserved Function of lincRNAs in Vertebrate Embryonic Development despite Rapid Sequence Evolution,» *Cell,* n. 147, p. 1537–1550, 2011.

[14] Bauer M, «Structural Alignment of Two RNA Sequences with Lagrangian Relaxation,» *Algorithms and Computation,* vol. 3341, pp. 113-123, 2004.

[15] I.Hofacker, «The Vienna RNA Secondary Structure Server,» 2003.

[16] F.Meyer, S.Kurtz, R.Backofen, S.Will and M.Beckstette, «Structator: fast index-based search for RNA sequence-structure patterns,» *BMC Bioinformatics,* 2011.

[17] T.Nagano and P.Fraser, «No-Nonsense Functions for Long Noncoding RNAs,» *Cell,* n. 145, pp. 178-181, 2011.

[18] Wang KC and Chang HY, «Molecular mechanisms of long noncoding RNAs,» *Mol Cell,* vol. 43, pp. 904-914, 2011.

[19] M.Tsai, O.Manor, Y.Wan, N.Mosammaparast, J.K.Wang, F.Lan, Y.Shi, E.Segal and H.Y.Chang, «Long Noncoding RNA as Modular Scaffold of Histone Modification Complexes,» *Science,* vol. 329, pp. 689-693, 2010.

[20] Rinn JL, Kertesz M, Wang JK et al. «Functional demarcation of active and silent chromatin domains in human HOX loci by noncoding RNAs,» *Cell,* n. 129, pp. 1311-23, 2007.