Journal of Molecular Graphics and Modelling 27 (2008) 299–308

Contents lists available at ScienceDirect

Journal of Molecular Graphics and Modelling

journal homepage: www.elsevier.com/locate/JMGM

Computational strategies for the automated design of RNA nanoscale structures from building blocks using NanoTiler☆

Eckart Bindewald [a], Calvin Grunewald [b], Brett Boyle [b], Mary O'Connor [b], Bruce A. Shapiro [b,*]

[a] Basic Research Program, SAIC-Frederick, Inc., NCI-Frederick, Frederick, MD 21702, USA
[b] Center for Cancer Research Nanobiology Program, NCI-Frederick, Frederick, MD 21702, USA

ARTICLE INFO

Article history:
Received 7 April 2008
Accepted 19 May 2008
Available online 24 May 2008

Keywords:
RNA
Design
Nanotechnology
Building block
Topology
Ring closure

ABSTRACT

One approach to designing RNA nanoscale structures is to use known RNA structural motifs such as junctions, kissing loops or bulges and to construct a molecular model by connecting these building blocks with helical struts. We previously developed an algorithm for detecting internal loops, junctions and kissing loops in RNA structures.

Here we present algorithms for automating or assisting many of the steps that are involved in creating RNA structures from building blocks: (1) assembling building blocks into nanostructures using either a combinatorial search or constraint satisfaction; (2) optimizing RNA 3D ring structures to improve ring closure; (3) sequence optimisation; (4) creating a unique non-degenerate RNA topology descriptor. This effectively creates a computational pipeline for generating molecular models of RNA nanostructures and more specifically RNA ring structures with optimized sequences from RNA building blocks. We show several examples of how the algorithms can be utilized to generate RNA tecto-shapes.

Published by Elsevier B.V.

1. Background

It has been shown that RNA sequences can be designed to self-assemble into a variety of three-dimensional structures, resembling squares, ladders or grids [1,2]. The design approach for these structures was to use elements of known RNA structures (like kissing loops or corner elements) and to connect these molecular building blocks with "struts" consisting of double helices. This approach is rooted in the observation that RNA structure is modular [3–5]. Computational tools for the analysis, prediction and design of RNA structures have been developed that work on the primary-, secondary- or tertiary-structure level [6–8]. There are many analogies between RNA and the DNA design; because of this we also mention, for selected cases, computer programs for designing DNA structures and sequences [9–12].

3DNA is a software package that allows analyzing, visualizing and rebuilding of nucleic acid tertiary structures [13]. NAB is a computer language for the description and automatic model generation of nucleic acid structures. It includes the use of distance geometry in order to build structures fulfilling distance constraints; the software also contains the AMBER force field, which facilitates the minimization and normal-mode analysis of generated models [14]. The program NAMOT is an interactive graphical software package for the generation and molecular modeling of nucleic acid structures [15]. Using a set of reduced coordinates, the program can alter a structure (by bending, stretching, compressing, etc.) while maintaining base pairing.

RNA2D3D [16] is a program for the fast generation of RNA 3D models from RNA secondary structure. It provides a wide spectrum of molecular modeling manipulations; among other things it can launch Tinker minimization and dynamics simulations from its graphical user interface. The software can also build exploratory tectoRNA structures.

MC-Sym is a program that can generate RNA 3D models that are compatible with constraints (such as base pairing interactions) [17]. It has been used for generating many published computational RNA models [18,19]. In combination with the MC-Fold program it was shown to be capable of predicting RNA secondary and tertiary structure with surprisingly high accuracy [20]. The program FARNA developed by Das and Baker [21] uses a fragment assembly approach in combination with a scoring function for *de novo* RNA tertiary structure prediction. The program S2S provides a

☆ The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.

* Corresponding author at: NCI-Frederick, Building 469, Room 150, Frederick, MD 21702, USA. Tel.: +1 301 846 5536; fax: +1 301 846 5598.
E-mail address: bshapiro@ncifcrf.gov (B.A. Shapiro).

1093-3263/$ – see front matter. Published by Elsevier B.V.
doi:10.1016/j.jmgm.2008.05.004

framework for integrating the tertiary structure of an RNA with sequence alignment information [22]. The MANIP software allows the rapid assembly of RNA motifs into larger three-dimensional structures using a graphical interface [23]. The program ERNA-3D is a molecular modeling program that allows one to interactively manipulate RNA 3D structures [24]. Specifically for DNA, a graphical molecular modeling program called Gideon has been developed [9]. The program TileSoft provides a graphical interface for designing DNA structures [10].

While the modeled tertiary structure is important for understanding properties of designed RNA structures as well as for applying atom-level simulation methods, it is the set of designed sequences that is potentially passed on to an experiment. Several programs have been developed to address the inverse secondary structure prediction problem (given an RNA (or DNA) secondary structure template, what set of sequences, if any, will fold accordingly?). In particular the programs RNAInverse [25], INFO-RNA [26,27] and RNA-SSD [28], or for DNA NANEV [11] and SEQUIN [12] have been developed for this purpose. Many methods for the prediction of secondary structures from a given sequence have been described elsewhere [6,7].

We previously developed the RNAJunction database which provides a large set of available RNA junctions, internal loops, bulges and kissing loops [29]. Building on this capability we present here a novel combinatorial approach for designing self-assembling RNA structures. In addition, we present several algorithms for the refinement of such combinatorially generated structures. Taken together, the algorithms form a computational pipeline for the automated generation of RNA nanostructures, in particular rings. The algorithms are implemented within the NanoTiler software. This program is a molecular modeling tool that among other things can use sets of RNA junction and kissing loop structures to generate RNA nanostructure models into which these building blocks can potentially self-assemble. While some of the ideas have been outlined in [8,30], we present here for the first time details of the algorithms and provide several examples of how this new approach can be used to generate novel structures from building blocks.

It should be mentioned that the NanoTiler software possesses or has in development many functionalities not mentioned here due to space and scope constraints. Examples are elastic network model capabilities, manual placement of structural elements and the automated placement of single- and double-stranded RNA fragments that bridge structural gaps.

2. Algorithms

In this section we describe the algorithms that constitute the computational pipeline for RNA ring and nanostructure design. The basic idea is to first obtain a set of candidate structures by performing a combinatorial search, trying out all allowed building block combinations and connectivities (algorithms A1 and A2). If the desired RNA structure contains cycles, these obtained candidate structures need to be refined by optimizing their ring closure (algorithms A3 and A4). Lastly the RNA sequences have to be fused (algorithm A5) and optimized (algorithms A6 and A7).

2.1. Topology classification

Using a combinatorial approach for the design of nucleic acid structures can lead to a large number of novel structural models that are only specified by their assembly rules and not by their final shape. How can one classify, compare and ''manage'' the plethora of structures generated in this fashion? One approach is to abstract the problem of nucleic acid structure comparison to the problem of comparing graphs (graph matching). Our approach is to represent RNA structural elements (junctions, kissing loops, internal loops, etc.) as graph vertices, the connecting helices as graph edges. Similar structure-to-graph mappings have been used by others ([31–33], however we are using a rigorous method [34] in order to map an RNA topology graph to a unique descriptor. Faulon et al. show that their topological descriptor is (unlike many other topological descriptors proposed in the literature) unique in the sense that two graphs have the same descriptor if and only if they are isomorphic. In order to use the computational chemistry software described in Ref. [34] without changes, we map RNA building blocks to pseudo-atoms and RNA helices to covalent bonds connecting these pseudo-atoms. An example of the flow from a 3D structure to a topology descriptor is shown in Fig. 1. The theory behind the topology descriptor is described in Ref. [34]. Important for the use as RNA descriptors is the concept that the topology descriptors of two graphs are identical if and only if the graphs are equivalent.

2.2. Simulated self-assembly for known connectivity and helix lengths: algorithm A1

Let us assume one wants to assemble a complex RNA nanostructure from a set of motifs or building blocks, much like the paradigm described by Jaeger and Chworos [35]. Using a conventional molecular modeling approach, the user has to know in advance which positions of the building blocks and which helix connectivities will lead to the desired 3D structure. What if the user is interested in the question ''given these building blocks, what different types of structures can be achieved with self-assembly''? Because of this, we allow the user to simply specify the connectivity rules (like ''connect helix 2 of building block type 3 with helix 1 of building block type 1, using a spacer of 5 base pairs''), and simulate a geometry-based self-assembly process starting from a seed building block for a specified number of generations. The algorithm immediately removes placed building blocks that would lead to severe steric clashes. In addition, the method detects the formation of cyclic structures and, if desired, exports structures containing cycles such as ring structures.

We define building block connectivity as a set of building blocks and a set of helix connection rules. A helix connection rule specifies which two helices of two building blocks are to be connected as

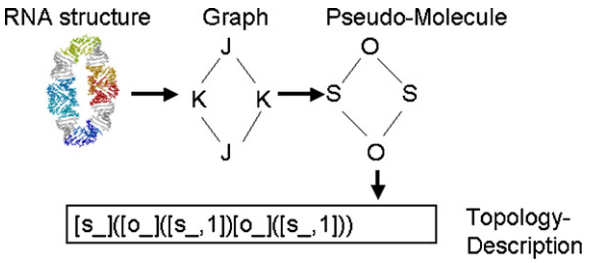

Fig. 1. Information flow of topology descriptor generation. We use an external computational chemistry program that computes a unique signature for a molecule [34]. In order to use this program, we use a mapping from building block type to a chemical element character (internal loops and bulges are mapped to ''O'', kissing loops are mapped to ''S'', three-way junctions are mapped to ''N'', four-way junctions are mapped to ''C''). The signature is in short a representation of a tree or trees originating from different atoms of the molecule. In this case the signature consists of one tree starting at ''S'' (represented as ''[s]''). Atoms appearing twice in the tree are indicated with a '',1'', in this case ''[s,1]''. The generation of the signature is described in detail in Ref. [34].