Automated Visualization of DNA Strand Displacement Systems

Malgorzata Nowicka and Vinay Gautam
Department of Computer Science, Aalto University, Finland
Emails: {malgorzata.nowicka, vinay.gautam}@aalto.fi

Abstract/Summary

DNA strand displacement (DSD) systems [1] have emerged as a widely-used methodology for embedding computational processes in the interactions of pre-designed short DNA strands. The DSD systems design methodology has been used to construct a variety of dynamic DNA systems, such as logic circuits, catalytic amplifiers and neural networks, as well as to emulate chemical reaction networks. Rendering non-overlapping 2D drawings of multi-stranded DNA structures is essential to understand interactions between different species of a given DSD system. Effective visualization of a DSD system is also helpful in communicating insights related to its modelling to a larger scientific community. In this work, we present a tool that produces the visualization of DSD systems at two levels. The species-level renders realistic, non-overlapping 2D drawings of multi-stranded DNA structures present in a given system. Species-level rendering includes structures with some of the most common pseudoknotted motifs whose planar drawings are challenging to render due to overlapping structural elements. The system-level rendering produces a network of enumerated reactions from a given DSD model that can be interactively visualized to gain quick insights about different species, their interactions and networks. The visualization tool is implemented in Python, integrated with the rule-based DSD reactions enumeration tool available at https://github.com/ashleylst/DSDPy.

1. Visualization Methodology

The basic terminologies, e.g. domain-level description of DNA structures, enumeration of DNA species and generation of a netlist of the enumerated set of species, used in our visualization methodology come from our earlier work on rule-based DSD modeling [2]. The rule-based methodology for modelling DSD systems uses a graph formalism in which DNA species are first represented as graphical structures that are then processed to generate new DNA species by applying graph rewriting rules based on a set of generic DSD reaction types. This graph processing exhaustively enumerates the entire space of DNA species reachable from the user's initial strand system design and then produces a list of DNA species and their DSD reaction network.

In the following, we briefly discuss a simple yet effective methodology used for an automated rendering and display of reactions networks and DNA species present in DSD systems. The visualization methodology is designed as a stand-alone Python module and provides an interface to the 'DSDPy' software pipeline. From the textual descriptions generated by DSDPy processor, 2D drawings of individual DNA species and reaction networks are produced.

1.1 Domain-level Visualization of Multi-stranded DNA Structures

Considering an example of two multi-stranded DNA species whose domain-level textual descriptions automated rendered drawings are shown in Figure 1. In the textual description, complementary domains are marked by (*) and bound domains are represented by named-pairing (!n, where n is unique label for each pairing). Our visualization methodology uses domain-level descriptions of DNA species, and we represent constituent DNA strands as strings of connected domains with hinges in between, where straight-line segments are used to represent domains. The initial configuration of DNA strings is created by placing the constituent strings along with the chords of a circle. The ultimate goal is to bring paired domains closer while separating apart unpaired domains to produce a non-overlapping depiction of the DNA structure in 2D.

Once we have an initial configuration of DNA strings, the system is perturbed by randomly selecting a string and moving one of its hinges at a time by translation or rotation. Since hinges connect the domains, the move is propagated to the adjacent domains. We define an *optimal drawing* of the structure as a non-crossing arrangement of the DNA-strings, where the paired domains are anti-parallel to each other, .i.e distance between the paired domains is minimal, and the angle between adjacent domains within the strand is maximal. Using a simple optimization function based on these optimization criteria, we used a simulated annealing algorithm [3] to produce realistic 2D drawings of multi-stranded DNA structures with and without pseudoknots.

Since there can be multiple arrangements of DNA strings around a circle, it is important that the strings' initial configuration is non-crossing to avoid the entangling of strings. In the case of a pseudoknot-free multi-stranded DNA structure, since there is always at least one such non-crossing arrangement, we find it out algorithmically and produce an initial configuration of DNA strings using this information.

#Domain-level description of S1
<X*!1 A*!2>
<A!2>
<L T2!3 X*!4 T1!5>
<A X!4 T2*!3>
<T1*!5 X!1 R>

#Domain-level description of S2
<a!1 b!2>
<b*!2 c d e!3>
<e*!3 f!4 a*!1 g f*!4>

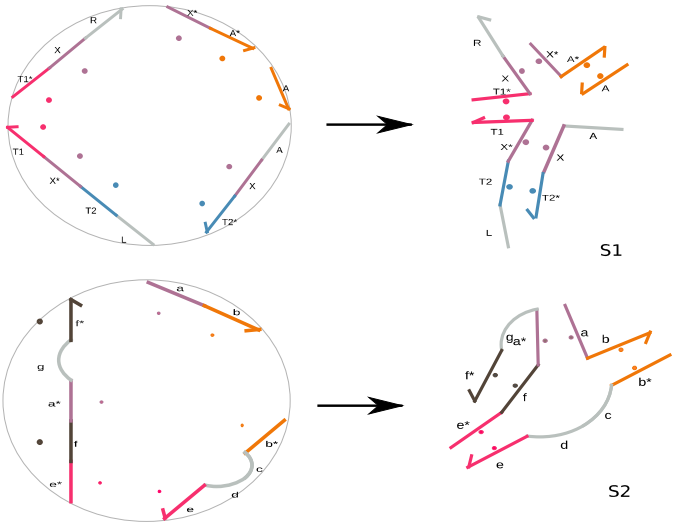

Fig.1: Automated visualization of multi-stranded DNA structures. Domain-level descriptions of two multi-stranded DNA systems (Left), and initial arrangements of strand systems and their respective rendered 2D drawings (right).

In the case of a pseudoknotted DNA structure, such a non-crossing arrangement of DNA strings does not exist. Our original approach for drawing a pseudoknotted DNA structure is based on algorithmically searching for the arrangements with the least number of crossings and then using a set of tricks of flipping DNA strings and domains to find an alternative arrangement free from crossing eventually. One can see that such an ad hoc approach is non-deterministic, and therefore, we do not always succeed in producing a non-overlapping drawing of pseudoknotted DNA structures. Creating realistic drawings of pseudoknotted DNA structures is computationally more difficult than drawing pseudoknot-free structures because a pseudoknotted structure is a graph (and possibly a nonplanar graph) with inner cycles within the pseudoknot and possibly outer cycles formed between the pseudoknot and other structural elements. One approach to address the planarity issue in the process of generating a 2D drawing of a pseudoknotted structure is to construct a straight-line representation of planar embedding of the given graph, for example, by using the edge addition method of Boyer and Myrvold [4] and straight-line drawing algorithm by Chrobak [5]. This approach has shown promising results in our recent efforts to algorithmically render a realistic drawing of the most commonly used pseudoknotted DNA structures in the DSD systems.

1.2 Visualization of DSD reactions network

A netlist of enumerated reactions obtained from the DSDPy processor provides the basis for visualizing the DSD reactions network. The tool supports multiple network views at different layouts, and parts of the network can further be interactively explored to quickly get insights about the interactions between different DNA species generated from the model.

References

[1] Zhang et al.,"Dynamic DNA nanotechnology using strand-displacement reactions". Nature Chemistry(2011).
[2] Gautam et al., "RuleDSD: a rule-based modelling and simulation tool for DNA strand displacement systems. In: 11th International Conference on Bioinformatics Models, Methods and Algorithms (2020).
[3] Kirkpatrick et al., Optimization by simulated annealing. *Science*, *220*(4598), 671-680 (1983).
[4] Boyer et al., „On the Cutting Edge: Simplified O(n) Planarity by Edge Addition.”Journal Of Graph Algorithms And Applications, 8(3) (2004).
[5] Chrobak et al., “Convex Grid Drawings of 3-connected Planar Graphs.” International Journal of Computational Geometry and Applications 7(3) (1997).