DNAjig: A New Approach for Building DNA Nanostructures

Md. Muksitul Haque[1], Ananth Kalyanaraman[1], Amit Dhingra[2], Nehal Abu-Lail[3], Kaori Graybeal[3]
[1]*School of Electrical Engineering and Computer Science*
[2]*Dept. Horticulture & Landscape Architecture*
[3]*Dept. Chemical Engineering & Bioengineering*
Washington State University
Pullman, WA, USA
Email: {*mhaque,ananth*}*@eecs.wsu.edu,* {*adhingra,nehal,kaori_graybeal*}*@wsu.edu*

***Abstract*—DNA self-assembly is an emerging technique in DNA nanotechnology that holds promise for high impact applications such as in the synthesis of DNA-based nanodevices in medicine, robotics and electronics. Recent advancements in technologies and ideas have spurred new growth in the area. The most significant challenge faced by designers in the field is the lack of algorithmic and software options to aid in the design process, and as a result the scope of synthesis has been restricted to modeling a limited set of shapes in 2D. In this paper, we propose a new design methodology called *DNAjig* to build DNA nanostructures. The highlights of this method are as follows: i) The construction procedure is based on a novel application of space-filling curves to model the shape of an arbitrary user-specified 2D or 3D object. ii) The method results in a simple, yet recursively constructable design layout that is inherently interlocked. iii) Almost all steps within the proposed design procedure can be automated and we present algorithms and a base-version implementation for the same. Wetlab validation showing the results of self-assembly of our first batch of computer generated 2D models is presented.**

***Keywords*-DNA nanosynthesis, computational modeling, Hilbert space filling curve, synthetic biology.**

I. Introduction

Self-assembly is a bottom-up construction process wherein a set of disordered particles combine to form a well-defined structure or shape as the result of specific interactions among themselves, without any external direction [15]. If DNA molecules are used as the building block during self-assembly, the process is called *DNA self-assembly* [16]. DNA self-assembly is guided by the nucleotide base pairing rule generated by the Watson-Crick complementarity between the two strands of a DNA molecule ($A \Leftrightarrow T$, $G \Leftrightarrow C$). The DNA molecule has several desirable attributes for use in nanotechnology. It has a minuscule diameter ($\sim 2nm$) and ∼3.66 bases for every nanometer linear length. Compared to RNA and protein, DNA molecules are highly stable and their sequence-dependent properties render predictability during material synthesis [6]. DNA self-assembly is expected to have a profound scientific and technological impact in the development of DNA-based nanomedicine, nanorobotics, DNA computing and DNA-based control for polymer and spatial crystal syntheses [12], [13].

While DNA self-assembly has been used to build crystals and arrays of limited dimensions before [3], [16], [17], a construction technique recently introduced by Paul Rothemund [9] scaled up the assemblage by orders of magnitude (from $\times 10^2$ nucleotides to $\sim 1.5 \times 10^4$). In this novel technique called "DNA origami", a single stranded DNA molecule (called the *template DNA*) is mixed with numerous fixed-length, short, single-stranded DNA fragments (called *oligos*) to result in the self-assembly of a desired 2D shape:

1) First, a 2D layout (to scale) is determined for the template DNA strand as if to "fill" the area covered by the target shape. This is achieved by first defining a seam position that divides the fill area into near-symmetrical halves, and then stacking up layers of the template DNA on either side of the seam.
2) Next, based on the template's folding path, oligos are designed for catering to two needs during self-assembly: i) their nucleotides collectively provide the complementary bases required to complete the second strand of the template DNA, and ii) they also play a structural role by "stapling" adjacent layers of the template strand eventually forcing it to take the desired shape.

While the above described design method has been successfully applied to yield stable nanostructures, there are *significant shortcomings* that a designer in the field currently faces. Firstly, the current method does not readily generalize to arbitrary shapes or into 3D. This is because with the irregularities in target shape it may get difficult, if not impossible, to find a suitable seam position[1] and/or an experimentally feasible layout for the template strand. Consequently, there are no software options to automate the design process for arbitrary shapes. In fact, the layouts for the original experiments [8], [9] and a few others that have followed [1], [5] were designed manually on an *ad hoc* case-to-case basis. Tools such as Nanorex [14] and GIDEON [2] support excellent interactive DNA editing features suited for molecular-level adjustments but lack automated design capability. From a practical standpoint, it should be easy to

[1] Seams were originally introduced as a matter of design convenience and do not particularly serve any structural purpose [8].

978-0-7695-3885-3/09 $26.00 © 2009 IEEE
DOI 10.1109/BIBM.2009.71

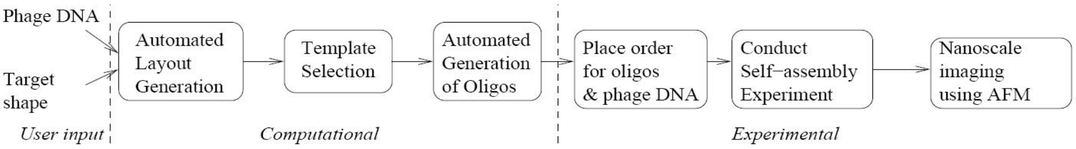
Phage DNA
Target shape
User input
Automated Layout Generation
Template Selection
Automated Generation of Oligos
Computational
Place order for oligos & phage DNA
Conduct Self–assembly Experiment
Nanoscale imaging using AFM
Experimental

Figure 1. The various phases involved in the DNAjig framework for DNA nanosynthesis.

see the utility of a fully automated design capability as it will be *necessary* to significantly enhance the experimental reach to arbitrary, more complex 2D and 3D nanostructures.

Another limitation of the current state-of-the-art is that there is no software assistance provided for the designer to select the region within the phage DNA for use as the template. The phage DNA is typically long (e.g., $7.2kb$ for M13mp18) and the template region should have some special sequence-level properties (e.g., specificity of oligo binding sites, sufficient G/C content). Scanning through every candidate region for these special properties can become cumbersome when performed manually.

A. Our Contributions

Here, we propose a new way to design DNA nanostructures. Our approach, called *DNAjig*, also uses the same experimental principles of self-assembly as used in DNA origami. However, the design layout of the template DNA strand and the oligo binding patterns generated are significantly different.

In short, we treat the problem as one of "filling" a polygon defined by the target shape and thread the template strand through a folding path that traverses the "pixels" covered by the polygonal area as guided by a *space-filling curve*[2]. Space-filling curves, popular in spatial data representation [11] generate deterministic, recursive patterns which is a feature of particular relevance to our application because such small, repeating patterns can be used as building blocks for constructing more complex, arbitrary shapes. The new construction procedure results in a relatively simple, elegant and structurally interlocked nanostructure. The design methodology is amenable to automation, and suited for modeling both regular and arbitrary shapes. While only 2D designs are considered in this paper, our algorithms are extensible to 3D. We also designed an algorithm to select candidates for template regions from within a phage DNA based on experimentally desired properties. These algorithms are described in Section II. Wetlab validation showing the results of self-assembly of our first batch of computer generated 2D models is presented in Section III.

II. Algorithms for Designing DNA Nanostructures

Problem statement: Given a phage DNA and a user-specified shape, determine a layout for the phage DNA and a set of oligo nucleotides that are experimentally expected to self-assemble into the target shape.

Figure 1 shows the different phases involved in our approach.

A. Automated Layout Generation

The challenges in determining a layout for the template DNA could be understood as follows. From a modeling perspective, the user-specified target shape can be said to represent a 2D/3D polygon with a certain pixel resolution, such that the template strand can be assumed to contribute a fixed number of nucleotides per pixel while self-assembling. A folding path should route the template strand through all and only those pixels within the polygon, visiting each pixel exactly once. This is the Hamiltonian Path problem, which is known to be NP-complete even under restrictions of a planar graph [4]. We use the Hilbert space-filling curve [10] for approximating the trajectory of the template DNA because of several desirable properties including its capability to generate a structure that a) preserves locality while threading the DNA through the polygonal area and b) appears inherently interlocked. In addition, the construction algorithm immediately carries over to arbitrary dimensions.

Squares: We start by describing the DNAjig designs for a square, and then extend the strategy for arbitrary shapes. Figures 2 shows our DNAjig designs for 2x2, 4x4 and 8x8 squares. The red lines show the Hilbert curves — same as the path of the template strand. A unit cell can hold up to a fixed number, ℓ, of nucleotides from the template DNA. The parameter ℓ is set to the minimum length of an oligo ($16b$ in our experiments). The design process is recursive, and is built out of 2x2 blocks (as the base case). We define four types of oligos, based on their location on the template and color code them for ease of exposition:

- **Blue:** These bind internally to cross-link the two arms of the U-shape.
- **Pink:** These bind to the base of the U-shape and cross-link two 2×2 squares.
- **Green:** These complement the bases in the midsection of a Hilbert curve where cross-linking is not possible.

[2] Space-filling curves are also used to construct jigsaw puzzles of arbitrary shapes and the name *DNAjig* is intended to reflect this analogy.

- **Purple:** These bind to the template segments that connect two neighboring squares, and may or may not cross-link depending on their Hilbert neighborhood (see Figures 2c for examples of both kinds).

All oligos have either $2 \times \ell$ or ℓ bases depending on whether they cross-link or not, respectively. It can be verified that these four oligo types would suffice for modeling arbitrary sized squares in 2D. The algorithm for design automation is as follows[3]:

S1) Using 2D matrix representation, construct the Hilbert curve and number the cells in the order in which they are visited. This labeling also corresponds to the nucleotide base position for the template.

S2) The oligos passing through each cell and their direction are determined by scanning through the Hilbert curve's traversal patterns adjacent to that cell.

Arbitrary shapes: Next, we describe DNAjig designs for arbitrary polygons and shapes (see Figures 3a-d). Our algorithm first finds the smallest bounding square for the target polygon such that the length along each dimension is a power of two. It then draws the Hilbert curve for the entire square. Along the Hilbert curve's traversal, some points will fall inside the polygon and the remaining outside. Ideally, the goal should be to traverse exactly once all and only those cells that fall within the polygon. Following the Hilbert curve inside the polygon can provide a direction to such a path (if one exists), or (if not) help enumerate segments of disjoint paths that can be combined together by either including cells immediately outside of the polygonal boundary or excluding a subset of cells inside. Subsequently, we will design oligos only for the template segments that best approximate the polygonal area. Assume that the cells inside the bounding square are numbered from 1 through n along the Hilbert curve from the top-right corner to the top-left corner. The algorithmic steps are as follows:

S1) Find the first polygonal cell through which the Hilbert curve first enters the polygon and stays inside for at least one more cell. This cell will be treated as the start of the template strand.

S2) Thereafter, from each cell u route the template strand to cell $u + 1$ provided $u + 1$ is also inside the polygon. If not, then locate the smallest cell v, such that $v > u + 1$ and v is inside the polygon. If no such v exists, then it means the filling has completed. Otherwise, compute the Euclidean distance in number of nucleotides, rounded to the nearest multiple ℓ, that the template strand should contribute to transit from u to v. This is represented by the dotted red line segments in Figures 3a and 3c. These dotted lines are inspected (currently done manually) and a decision is made on the whether to design oligos for those segments or leave them single-stranded to facilitate subsequent removal using a single-strand digestion enzyme post self-assembly.

S3) Once the template strand has been routed through the entire polygon, use the oligo placement scheme described for squares and fill each polygonal cell covered by the template using appropriate oligos.

B. Template Selection

Once a template folding path and oligo placements have been determined by the algorithms described in Section II-A, the next step is to select an appropriate region within the phage DNA that can act as the template. Once selected, the oligos will be designed to target that region within the whole genome. Therefore, there are a couple of properties a template region should ideally possess: i) **uniqueness:** The region should be as less repetitive as possible to ensure the specificity of binding sites for oligos; and ii) **G/C content:** The overall G/C content of the template DNA should be at least as much as a user-specified threshold.

Let L be the length of the template DNA and G denote the genome sequence of the phage. First we shortlist L-length substrings that uniquely occur in G by constructing a suffix tree [7] of G and examining paths that are uniquely represented at a tree-depth of L below the root. Next, we identify a subset of the shortlisted substrings which contain the most number of unique $\frac{\ell}{2}$-mers. This is because the oligos bind to the template in increments of $\frac{\ell}{2}$ nucleotides in our DNAjig design.

C. Automated Generation of Oligos

Once the nucleotide sequence of the template is determined, then determining the nucleotide sequence of the oligos of length ℓ is straightforward and is achieved by mapping the oligo placements back to the template sequence and reverse-complementing the bases. These oligos are guaranteed to map to a contiguous stretch (substring of length ℓ) within the template. For oligos which are of length $2 \times \ell$, their prefix and suffix of length $\frac{\ell}{2}$ will map to the template DNA in two different locations, and therefore their nucleotide sequences can be directly deduced by reverse-complementing the corresponding template DNA segments. As for the oligo's midsection, we are free to determine its sequence as the nucleotides here do not bind with the template DNA. Therefore, we generate a random sequence of length ℓ with sufficient G/C content. Once assigned, the midsection of the other oligo that binds to this oligo is automatically computed using another reverse complement operation.

Implementation: All of our code is implemented using C and the GIMP Toolkit (GTK). Figure 3e shows a snapshot of one of our designs generated by our software.

III. Experimental Results

To experimentally validate our DNAjig design templates, we conducted wetlab experiments to generate the self-

[3] Finer details of our implementation are omitted due to lack of space.

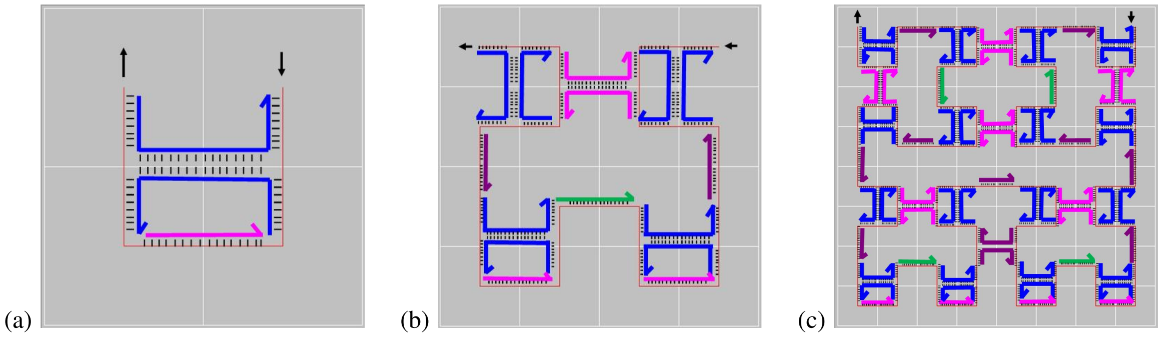

Figure 2. Parts (a), (b) and (c) DNAjig designs for 2x2, 4x4 and 8x8 squares, respectively. Shown are their respective placement strategies for oligos (identified by different colors) over the template strand (identified by the red line). Arrow heads point to strand directionality from $5' - 3'$. The base-to-base pairing between the template and oligos, and between oligos are shown in small black lines.

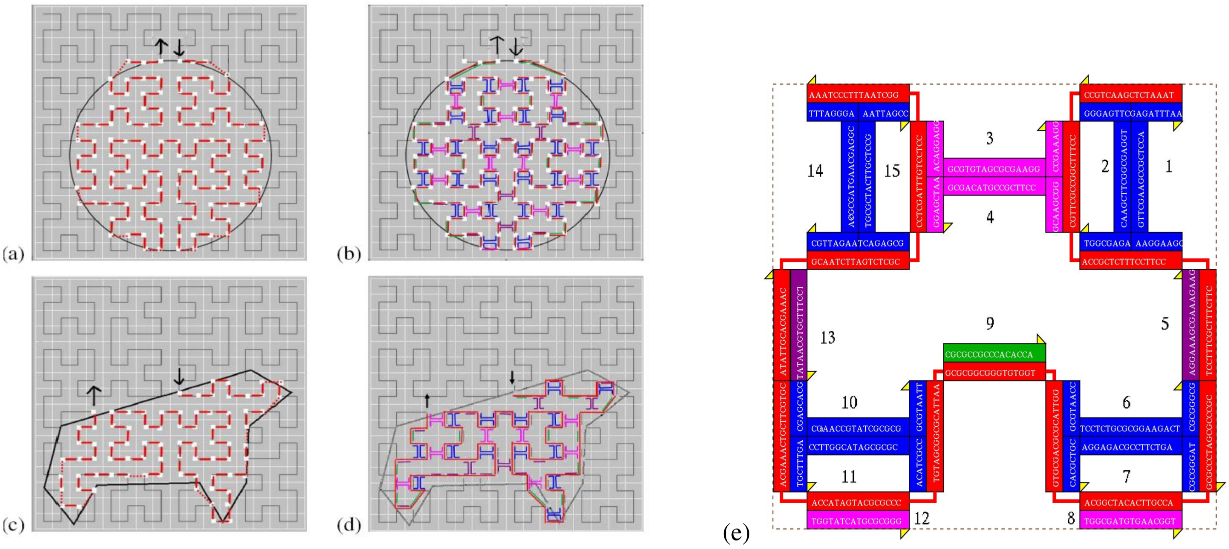

Figure 3. Parts (a)(b)(c)(d): DNAjig designs for a circle and a polygon. Parts (a) and (c) show the template path computed by the algorithm. The dotted red lines show the result of bypassing the Hilbert curve path outside the polygonal area. Parts (b) and (d) show corresponding DNAjig designs complete with oligo placements. Part (e): A snapshot from our software showing a software-generated DNAjig design for a 4x4 square complete with sequence information and orientation. The red segments denote the template strand and the oligos, which are numbered from 1 through 15, are denoted by other colors. Arrow heads are placed at $3'$ ends.

assembly of an 8×8 square (Figure 2c) with an estimated dimension of $\sim 55nm \times 55nm$. Four separate experiments were designed to incrementally build the square by adding one quadrant per stage. This way, we were able to also test intermittent shapes — a 4×4 square, a 4×8 rectangle and a 3-quadrant $L-$ shape. The same experimental protocol was followed for each experiment: First, a single-stranded phage DNA of M13mp18 (7,250bp) was acquired from New England BioLabs. The DNA sequence and the corner coordinates of the target shape were input into the DNAjig software framework, which subsequently generated a template layout and the corresponding placements and sequence description of the oligo strands. The software generated a total of 70 oligos (22 16b-long and 48 32b-long) for the 4 experiments. Using the phage DNA and the oligo DNA sequences, self assembly was performed using the procedure described earlier [9]. An aliquot of 3 μl was used to test for self assembly using Agarose gel electrophoresis.

Agarose gel electrophoresis is a standard molecular biology method to separate DNA fragments based on their size. DNA is a negatively charged molecule and thus during electrophoresis DNA migrates towards the positive electrode. Electrophoresis is accomplished in electrophoresis units with negative and positive electrodes at the N and S end. Agarose gel is constituted by boiling agarose at 1% concentration in 1x TAE (Tris-Acetate-EDTA) buffer. The gel is submerged in 1X TAE buffer that serves the purpose of conducting electricity through the agarose gel. DNA solution is mixed

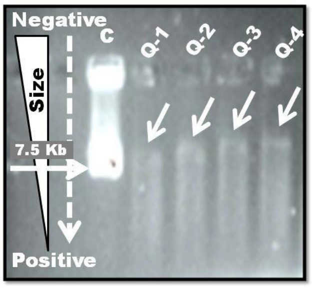

Figure 4. Agarose gel electrophoresis was used to confirm incremental self-assembly of the M13mp18 DNA 7 kb single stranded DNA template (7250 bp in size). Lane C: M13mp18DNA template; Lane Q-1: DNA template self assembled with Quadrant 1 primers (NE); Lane Q-2: DNA template assembled with Quadrant 1-2 primers (NE + NW); Lane Q-3: DNA template assembled with Quadrant 1-3 primers (NE + NW+ SE); Lane Q-4: DNA template assembled with Quadrant 1-4 primers (NE + NW + SE + SW) . Arrows in lanes labeled Q1 to Q4 indicate progressive increase in size of DNA template due to increased secondary structure formation as a result of self assembly. Lane C contains 500 ng of DNA. Lanes Q-1 to Q-4 contain approximately 50 ng of DNA thus, the difference in observed intensity of the DNA loaded. DNA migrates towards the positive electrode with the smaller fragments migrating faster.

with a loading buffer containing 30% glycerol in wells at the N end. Electricity is applied at 50 V for variable amount of time to separate the DNA fragments. Agarose gel consists of a matrix through which the DNA molecules migrate. Smaller DNA molecules will travel faster. By varying the concentration of agarose the pore size of the agarose matrix can be changed. Separation of DNA is a property of its size and conformation. In this experiment we used the latter property to observe self assembly.

The agarose gel electrophoresis results are shown in Figure 4. Size of the control fragment is $7.5kb$ loaded in lane C. Over 500 ng of DNA is loaded thus it appears higher in intensity. Lanes Q1 to Q4 represent the self assembled template. As is clearly visible and marked with arrows, there is an incremental increase in the size of the DNA fragments as each quadrant is added to the self assembly reaction. The conformation of DNA changes with the oligos annealing at specific locations which has manifested itself as increased size during agarose gel electrophoresis. Visualization of our self-assembly products is currently underway using Atomic Force Microscopy, following standard experimental protocols described earlier [9].

IV. Conclusions

Designing experimentally stable 2D and 3D DNA nanostructures is of paramount importance in DNA nanotechnology with a wide range of important applications. Yet, significant challenges remain in designing arbitrary shapes and in modeling 3D objects. In this paper, we proposed a new method called *DNAjig* for building DNA nanostructures. DNAjig exploits various features of the Hilbert space-filling curve to support automated building of interlocked nanostructures for arbitrary user-specified shapes. The design methodology facilitates design automation as demonstrated through the algorithms presented. This is an important step forward in enabling the design of arbitrary complex nanostructures. Important extensions remain to be investigated as part of our future development plans.

Acknowledgments

We are grateful to Paul Rothemund for helping us in the understanding of DNA self-assembly and origami technique via constructive discussions and email exchanges. Hilbert curve images were generated using the applet at http://www.cs.utexas.edu/users/vbb/misc/sfc/0index.html.

References

[1] E.S. Andersen *et al.* DNA origami design of dolphin-shaped structures with flexible tails. *ACS Nano*, 2(6):1213–1218, 2008.

[2] J.J. Birac *et al.* Architecture with GIDEON, a program for design in structural DNA nanotechnology. *Journal of Molecular Graphics and Modelling*, 25:470–480, 2006.

[3] S.H. Park *et al.* Finite-size, fully-addressable DNA tile lattices formed by hierarchical assembly procedures. *Agnew. Chem.*, 350(631–633), 2006.

[4] M.R. Garey *et al.* The planar Hamiltonian circuit problem is NP-complete. *SIAM Journal of Computing*, 5(4):704–714, 1976.

[5] Y. Ke *et al.* Self-assembled water-soluble nucleic acid probe tiles for label-free RNA hybridization assays. *Science*, 319:180–183, 2008.

[6] Y. Lu and J. Liu. Functional DNA nanotechnology: emerging applications of DNAzymes and aptamers. *Current Opinion in Biotechnology*, 17:580–588, 2006.

[7] E. McCreight. A space economical suffix tree construction algorithm. *Journal of the ACM*, 23:262–272, 1976.

[8] P. Rothemund. Personal communication. 2009.

[9] P.W.K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440:297–302, 2006.

[10] H. Sagan. Chapter 2: Hilbert's space-filling curve. In *Space-filling curves*. Springer, 1994.

[11] S. Seal and S. Aluru. Chapter 44: Spatial domain decomposition methods for parallel scientific computing. In *Handbook of Parallel Computing: Models, Algorithms and Applications, (Ed. S. Rajasekaran and J. Reif)*. Chapman & Hall/CRC Computer and Information Science Series, 2007.

[12] N.C. Seeman. An overview of structural DNA nanotechnology. *Molecular Biotechnology*, 37:246–257, 2007.

[13] F.C. Simmel. Towards biomedical applications for nucleic acid nanodevices. *Nanomedicine*, 2:817–830, 2007.

[14] M. Sims. *Nanorex. http://www.nanoengineer-1.com*, 2008.

[15] G.M. Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295:2418–2421, 2002.

[16] E. Winfree. Algorithmic self-assembly of DNA: Theoretical motivations and 2D assembly experiments. *Journal of Biomolecular Structure & Dynamics Conversations*, 11(2):263–270, 2000.

[17] Y. Zhang and N.C. Seeman. The construction of a DNA truncated octahedron. *J. Am. Chem. Soc.*, 116:1661–1669, 1994.