TileSoft: Sequence Optimization Software for Designing DNA Secondary Structures

P. Yin*, B. Guo*, C. Belmore*, W. Palmeri*, E. Winfree†, T. H. LaBean* and J. H. Reif*

* Department of Computer Science, Duke University
† Department of Computer Science, Caltech
Abstract

DNA is a crucial construction material for molecular scale objects with nano-scale features. Diverse synthetic DNA objects hold great potential for applications such as nano-fabrication, nano-robotics, nano-computing, and nano-electronics. The construction of DNA objects is generally carried out via self-assembly. During self-assembly, DNA strands are guided by their sequence information into secondary structures to maximize Watson-Crick pairing of their bases and thus minimize the free energy of the resultant structures. A crucial computational problem in constructing DNA objects is the design of DNA sequences that can correctly assemble into desired DNA secondary structures. However, existing software packages only provide unintuitive text-line interfaces and generally require the user to step through the entire sequence selection process, which could be time-consuming and tedious. In contrast, TileSoft described here deliver the following features:

• Its **graphical user interface** renders the molecular architect the ability to define DNA secondary structure and accompanying designing constraints directly on the interface as well as the ability to view the optimized sequence information pictorially.

• Its **fully automatic optimization module**, which uses an evolutionary algorithm built on top of DNADesign by E. Winfree, relieves the user of the drudgery of manually dictating the sequence selection process, and its evolutionary algorithm produces satisfactory results efficiently.

• Its graphical user interface and its optimization module are **smoothly integrated** from user's perspective, while they are at the same time **well separated** in terms of software architecture, making each amenable to future improvements without negatively affecting the other.
Motivation: Designing DNA tiles

- DNA as nano-construction material
- Self-assembly as bottom-up nano-construction method
- DNA lattices made of DNA tiles, i.e. smaller DNA secondary nanostructure units
- Design process:
  1. Template design
  2. Sequence selection (optimization)

Tile template to be optimized
Prior work v.s. TileSoft

• **DNA word design software:**
  • Produce a pool of DNA sequences such that each sequence is of maximal difference from others

• **Sequin:**
  • Generate DNA sequences that uniquely assemble into desired secondary structures
  • Text line interface for inputting template and displaying optimized sequences
  • Sequence optimization process only semi-automated

  • Generate DNA sequences that uniquely assemble into desired secondary structures
  • Graphical interface for inputting template and displaying optimized structure (GUI Module)
  • Sequence optimization process fully automated (Optimization Module)
  • GUI Module and Optimization Module smoothly integrated for end users, yet well separated in software architecture
GUI: Default window
GUI: Define Crossover

• The user can define crossovers between helices, by clicking sequentially the two bases to be connected in 5' to 3' order.
GUI: Set 5’ end; set 3’ end

Set 5’ end

- By setting the 5' end and 3' end of a DNA strand, the user specifies the length of the strand, and the unused segment of the strand is deleted automatically (shown in color gray).
GUI: Edit base

- The user can directly input the base values for a strand by typing; Typing more than one character edits consecutive bases in the 5' to 3' direction along the strand.
GUI: Set non-Waston-Crick base pairing

- The user can define the subsequences that are not required to be Watson-Crick base paired by clicking on the starting and ending bases of the subsequences.
GUI: Set and show EQ constraint

Set EQ constraint:
- **Set EQ constraint**: Clicking on two bases in one strand defines the starting and ending points of the first sub-sequence, and a click on another base delineates the second sub-sequence with the same length and direction as the first one.

Show EQ constraint:
- **Show EQ Constraint**: A small window is brought up that contains multiple buttons, with each representing a set of equal sub-sequences. When one of these buttons is clicked, the corresponding sub-sequences will be highlighted in purple.
Optimization module

- The optimization module is developed based on *DNADesign* by E. Winfree.

- The optimization module enhances *DNADesign* employs an *evolutionary algorithm* to find the best solution to the optimization *objective function*.

### Evolutionary algorithm:
- An evolutionary algorithm maintains a population of DNA sequences, which are generated randomly during *initialization*. During *selection*, the fittest DNA sequences are chosen for reproduction, based on their score according to the *objective function*. These individuals are used to generate new individuals via *mutations* and *crossovers*, and the newly produced individuals are *reinserted* into the population. The process is repeated until meeting some *termination* condition.

### Objective function as in DNADesign:
- The objective function consists of two weighted factors, the count of unwanted complementary sequences *spurious matches* (as the sequence-symmetry minimization algorithm used in Sequin) and the count of *to-be-avoided sub-sequences*, e.g. long AT runs.
Future work

- **GUI:**
  - Geometrically more flexible structures
  - Make the number of helices and number of bases per helix user specifiable

- **Optimization:**
  - Multiple parallel traces of optimization process with different starting points
  - A pre-optimized library
  - Incorporate parameters such as hybridization temperature and software modules such as BIND
  - A new heuristic that performs optimization based on existing pre-optimized duplex libraries

- **Other:**
  - Curvature analyzer (S. H. Park, Duke Physics)
  - Make the software more robust