Testing-Optimized DNA Assembly Algorithms

Evan Appleton[1] Viktor Vasilev[1] Swapnil Bhatia[2] Traci Haddock[2] Douglas Densmore[1,2]
[1]Graduate Program in Bioinformatics
[2]Department of Electrical and Computer Engineering,
Boston University, 8 Saint Mary's St. Boston, MA, USA 02215
{eapple,vvasilev,swapnilb, thaddock,dougd}@bu.edu

1. MOTIVATION

Synthetic biologists formulate a desired cellular behavior and engineer cells to express it. One way to approach cellular engineering is parts-based, where biologists design a synthetic genetic regulatory network (GRN) of DNA parts to create that behavior and physically construct DNA to insert into cells. The synthetic biology community has focused upon improving DNA assembly methods - how to make it faster, cheaper and more reliable. Some popular assembly methods are binary and some are one-pot [3].

Recent work has been done to develop algorithms that optimize binary assembly [2]. These algorithms do not, however, account for what is being built or the utility of assembly intermediates. This is particularly problematic because each step of DNA assembly is prone to multiple sources of error. Frequent errors occur in PCR, ligation, and gel extraction, and furthermore, parts with correct sequence sometimes do not function as expected. Thus, it is advantageous to construct assembly intermediates that are easily testable and reusable. By accounting for the biological features of parts, an assembly plan can be devised that maximizes the formation of intermediates with the best biological qualities.

Based upon prior optimization algorithms [2], here we present algorithms that select an assembly plan optimized for structural and functional testability. Before calculating the optimal plan, an algorithm identifies the most common part-order motifs within a selected database to maximize future reusability of intermediates. These additions to the state-of-the-art can considerably change the optimal assembly plan, even for small parts (Figure 1).

2. STRUCTURAL TESTING

The structural model of testing in electrical engineering assumes that within a circuit, all parts have known function and function properly [1]. Structural tests are to ensure that all parts are properly connected and undamaged. In genetic engineering, we define structural tests as tests to determine that DNA parts have the correct structure (primary, secondary, etc.) as per their structural specification. A structural specification consists of a DNA part with defined structure (primary, secondary, tertiary and quaternary).

It is important that a final construct abides its specification, but because it is difficult to backtrack and replace small pieces during DNA assembly of a large construct, it is important that intermediate parts satisfy their structural specification, as well. Structural information can be acquired with DNA sequencing, which captures reliable primary structure information, but sequencing all intermediates is too costly

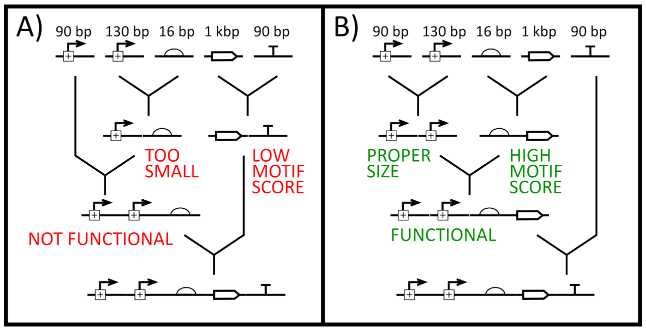

Figure 1: Assembly Plan Comparison A) State-of-the-art algorithms find an optimal assembly tree with intermediates that are too small, do not function and have low motif scores. B) The modified algorithm selects an assembly tree with the same score as the original algorithm but improves the intermediates to be the proper size, functionally testable and have a high motif score.

for large constructs. Another cheaper primary structure testing method, restriction mapping, is more practical for this purpose but imposes two compositional constraints to enable testing: (1) Parts of length $< 200bp$ are difficult to extract from a standard 1-1.5% agarose gel and (2) The part and plasmid vector must have the required restriction sites to allow for all bands to be distinguishable on a gel.

3. FUNCTIONAL TESTING

In contrast to the structural model, the functional model of testing does not assume that parts function properly. Rather, functional testing aims to validate that a circuit functions as per its functional specification [1]. Here we define a functional specification as a type-ordered DNA part that produces something in cellular context (i.e. mRNA, proteins, etc.). As per the specification example in Figure 2B, two proteins are produced that interact with promoters. In this example, we assume all three genes are fused with fluorescent protein genes, so all proteins are detectable by flow cytometry and that all assembly backbones have flanking terminators. Therefore, we define a functional unit as any part containing a Promoter-RBS-Gene part type, as in Figure 2C.

To maximize functional testability of intermediates, we bias the assembly plan towards constructing functional units as soon as possible. Once these testable functional units are built and pass a functional test, they will again be tested with the addition of each new part, as an incremental method

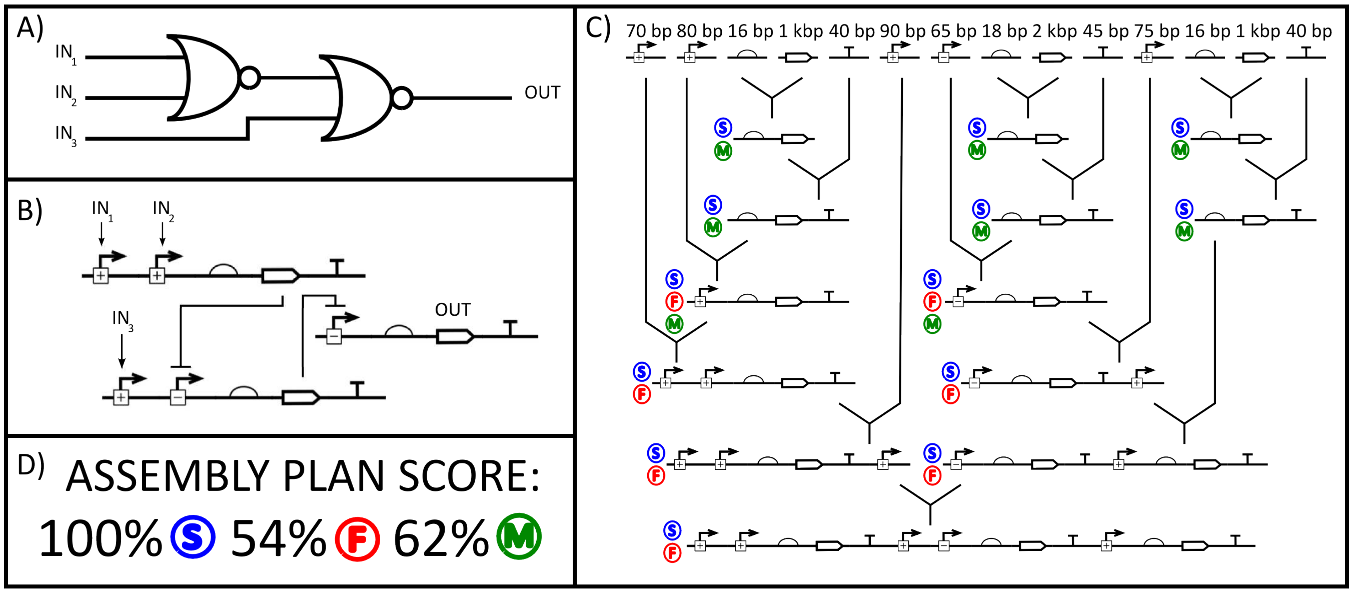
A)
IN1
IN2
IN3
OUT
B)
IN1
IN2
IN3
OUT
C)
70 bp 80 bp 16 bp 1 kbp 40 bp 90 bp 65 bp 18 bp 2 kbp 45 bp 75 bp 16 bp 1 kbp 40 bp
S
M
F
D)
ASSEMBLY PLAN SCORE:
100% S 54% F 62% M

Figure 2: Example Modified Assembly Plan. A) A cascading genetic NOR gate B) DNA construct to satisfy this specification. C) Assembly plan: motif training step finds that RBS-Gene, RBS-Gene-Term and Prom-RBS-Gene-Term are the highest scoring motifs (M) and any sequence containing Prom-RBS-Gene is functionally testable (F). D) The testability and high scoring motif-abundance will be reported as a percent coverage, resulting in an assembly plan score. Reported is this plan's structural testability score (S), functional testability score (F) and high motif score (M).

to track the composite part's function as more parts are added. This is favored because sometimes as parts are incrementally added, the original part's functionality changes. While incremental testing is good, selecting a fully incremental assembly plan requires too many assembly stages for large constructs.

4. MOTIF TRAINING

To maximize the reusability of intermediates, we analyze a part database to identify the most abundant ordered-part-type motifs (i.e. RBS-Gene as per Figure 2C) and assess them high motif scores. In calculating the assembly plan, stages, steps, testability, and abundance of high-scoring motifs factor into the selection. This step is aimed at identifying type-ordered intermediates that are the most commonly reused.

5. ALGORITHM

The algorithm accepts three parameters: GP (parts comprising the assembly plan), PL (parts already physically assembled), and TDB (a training dataset used for motif training). The algorithm finds all AGRN motifs in TDB to calculate reuse count. The more frequently used motifs are favored in new assemblies. We assume that motifs frequently used in the past will occur in future assemblies. Internal functions compare assembly graphs and select them based on criteria for structural and functional testability and/or re-occurring motifs. The provided algorithm pseudo-code provides a brief introduction to the process.

6. CONTRIBUTIONS

These additions to the state-of-the-art improve the utility of automated assembly planning by taking important biological features of DNA parts into consideration. Specifically,

Algorithm 1 Assembly for Test (goal parts, part lib, db)

```
$mhash = []$
$parts = FindAllPartsReuseCount(database)$
for all $comppart \in parts$ do
    $motif = []$
    for all $basicpart \in comppart$ do
        $motif.add(basicpart.type)$
    end for
    if $if(motif \exists \in mhash)$ then
        $mhash[motif] = 0$
    end if
    $mhash[motif] = mhash[motif] + comppart.rcnt$
end for
$mhash.weights = NormalizeMotifCount(mhash)$
return $Algorithm(goalparts, partlibrary, mhash)$
```

we define two testability specifications and methodology for optimizing the assembly plan for testing these specifications. This results in accelerated assembly of target constructs and also future constructs that use similar building blocks.

7. REFERENCES

[1] Abramovici, M., Breuer, M., and Friedman, A. *Digital Systems Testing and Testable Design.* AT&T, IEEE PRESS, Piscataway, NJ, USA, 1990.

[2] Densmore, D., Hsiau, T. H., Kittleson, J. T., DeLoache, W., Batten, C., and Anderson, J. C. Algorithms for automated DNA assembly. *Nucleic Acids Research 38*, 8 (2010).

[3] Weber, E., Engler, C., Gruetzner, R., Werner, S., and Marillonet, S. A modular cloning system for standardized assembly of multigene constructs. *PlosONE 6*, 2 (2010).