THE INTRALABORATORY TOOL FOR GOLDEN GATE ASSEMBLY AND RESTRICTION ANALYSIS

1. Introduction

Golden Gate (GG) assembly is a method of molecular cloning that provides the opportunity for simultaneous insertion of multiple fragments into a destination vector [1]. It relies on the activity of Type IIS restriction enzymes that cleave at a certain distance upstream of 5'-3' recognition site (or downstream of 3'-5' binding region). These enzymes create four bp-long overhangs ("sticky-ends"), the nucleotide content of which is dictated only by the sequence itself, thus providing 4^4 = 256 possible combinations [1-3].

The main goal of preparation for the GG cloning is to design primers that will introduce: the binding sites for Type II restrictase, four bp-long regions (that will eventually create sticky ends complementary to the next and previous sequences), as well as spacers regions (e.g., Kozak sequence, or GC-linker). While all these elements can be predicted by either known or coding sequences, the compatibility of sticky ends in different junctions of the assembly remains a problem that requires pragmatic trial rather than computer prediction [2,3]. In many cases, Smith-Waterman alignment fails to fit experimental data, since DNA ligases can join mismatched nucleotides with particular specificity (which in its turns depends on the experimental conditions, such as incubation period and temperature), and thus generate non-specific unpredicted ligation events [3].

The second practical problem that a researcher faces after the ligation step is choosing the type IIP restriction enzymes for the test digest (or restriction) analysis of the assembled construct. If in the classical cloning, one usually opts for endonucleases that were used for fragments digestion, in the GG assembly it is impossible, since no recognitions sites for Type IIS endonucleases will be observed in the final construct ("scarless cloning"). Available web-tools or programs (such as Benchling, ApE, and others) only allow for a manual combination of restriction enzymes (by selecting different enzymes and analyzing the bands they produce until the desired result is achieved) [4,5]. Such manual combinations are time-wasting and should be automated.

Overall, the current project aims to achieve optimization of the GG cloning efficiency by addressing two major practical problems – predicting the compatibility of the sticky-ends of different junctions in the one-pot reaction, and automatization of the choice of type IIP restriction enzymes for restriction analysis of the assembled plasmid.

2. Methods

In order to certainly predict the compatibility of the sticky-ends in the GG reaction, I used *practical* high-throughput data (Reference: [3], Supplementary materials → sb8b00333_si_002.zip → FileS03_T4_18h_25C.xlsx) for the sticky-end joining by T4 ligase at 25°C (the similar condition we use intra-laboratorially). This table provides a normalized score for ligation events between all the possible four bp-long oligonucleotides (the heatmap representation of the data is showed in Figure 1).

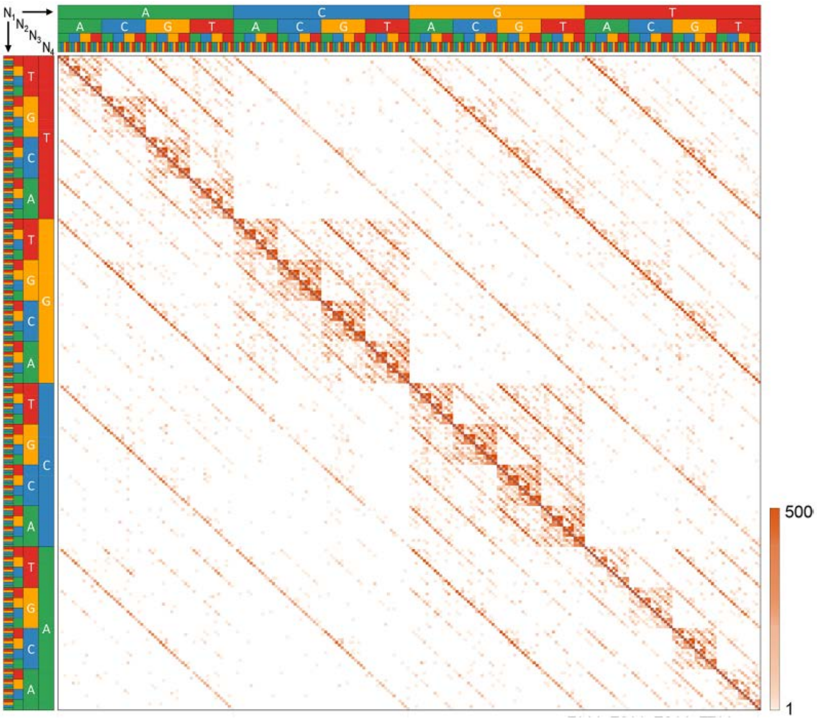


Figure 1 Frequency heat map (log-scaled) for the ligation of randomized four-base overhangs by T4 DNA ligase (18 h at 25 °C). Reference: *Potapov at al., 2018*

Thus, by checking all the pairwise combination of all four bp long overhangs in the reaction (pairwise comparison matrix) for non-specific ligation events according to the practical data, one can speculate about very high fidelity of the planned GG assembly (Please address the function "check_sticky.m" in the code repository for more details). *Potapov et al.* also have shown that such predicted fidelity reflects the actual data obtained in 10+ inserts GG assemblies.

For resolving the problem of enzyme choice for the restriction analysis, I applied a descriptive model for gel bands separation (Figure 2). The function f(distance) = $-20(1/(0.45x + 0.75)^8 - 1/(0.8x+0.5)^2)$ takes the maximal values at about 1 kb (1000 bp) distance between *two closest* bands on the gel (optimal separation) and either decreases rapidly if two bands are too close together (the left side of the graph) or slows down moderately if two closest bands are distantly separated (the right side) to enable the more extensive separation range. The threshold (horizontal line) serves to abolish the closest bands with the distance < 500 bp between them, as well as two closest, but very distantly (>5200 bp) separated bands (e.g., having two bands very 'heavy' and 'light' bands, such as 6000 and 700 bp, is not optimal for the visual result and time of separation). The presented range 0.4 – 6 kbs reflects the DNA fragments sizes capable of separation in 1-2 % agarose gel.

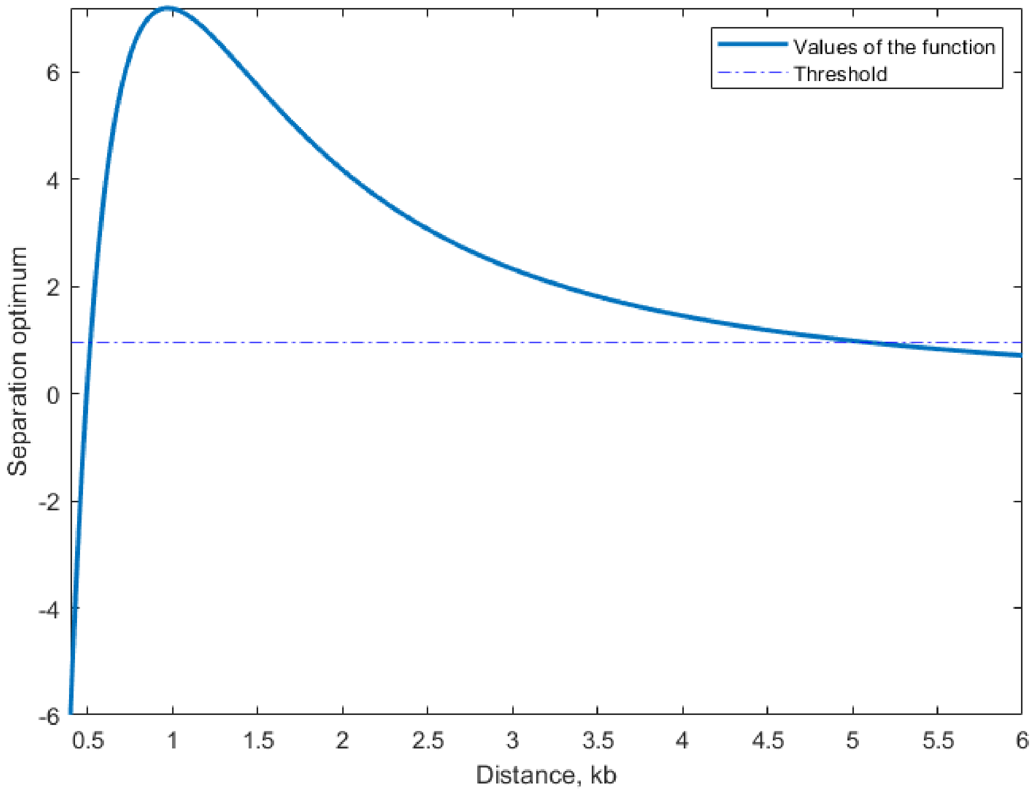


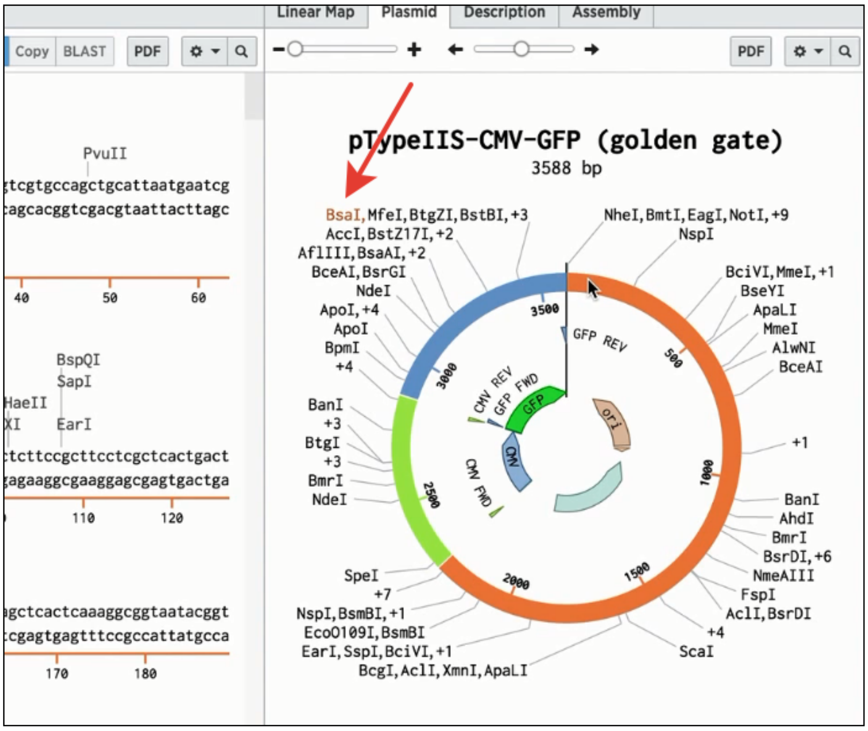
Figure 2 The descriptive model for the gel bands separation in 1-2% agarose gel. Please see the text main text and "*Supplementary_ test_digest.m*" in the code folder for more information.

3. Results

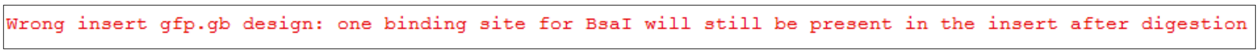
3.1. Created a tool for accurate Golden Gate assembly (*GG_assembly.m*)

One of the most popular tools for GG assembly is Benchling Assembly Wizard [4]. Although it is user-friendly and well-designed, it has at least two disadvantages in the execution part, which one can notice working with the tutorial sequences.

First, it may create an output assembly with the recognition sites for the IIS restriction endonuclease used in the GG reaction (Figure 3). Such reaction would lead to the daunting GG experimental results, since amplified GFP insert would be cut in 3 sites, creating non-specific sticky-ends and undesired coding sequences, the final correct plasmid in the reaction (if any possible) would be linearized by BsaI cutting.



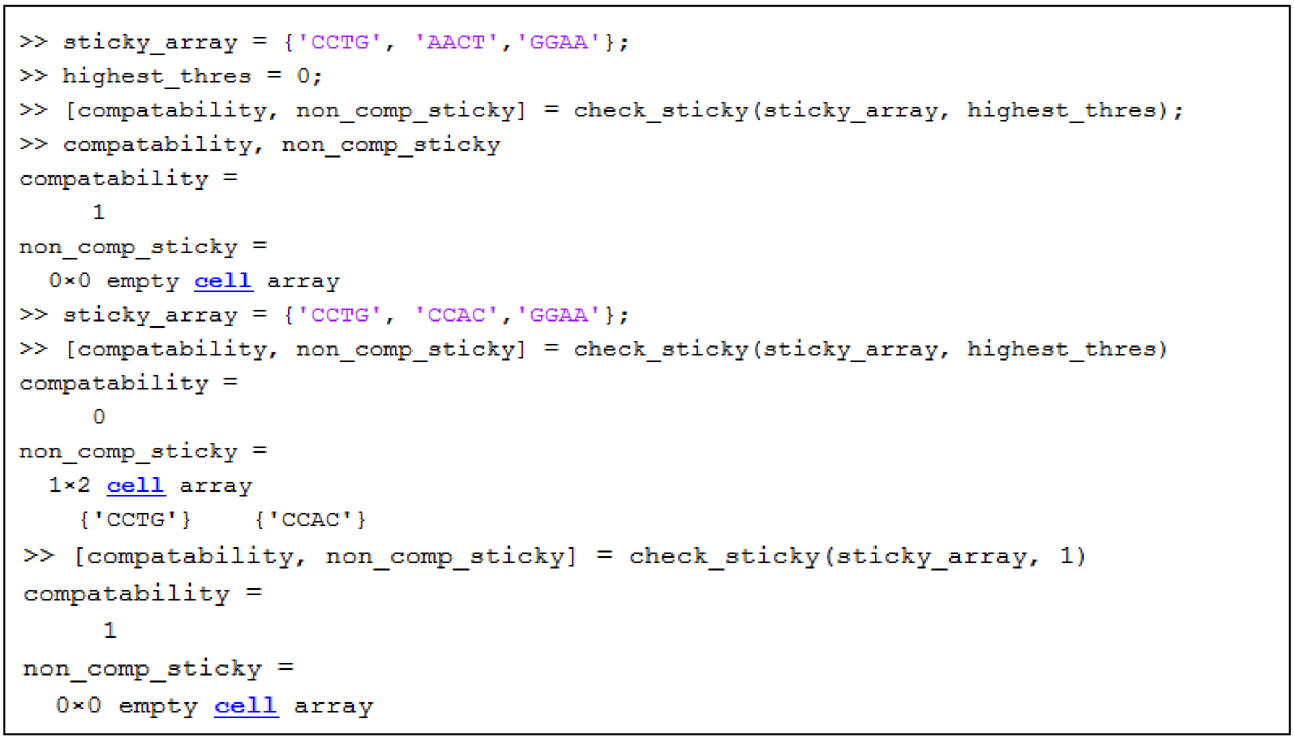
The desired notification (implemented in GG_assembly.m):



```
Wrong insert gfp.gb design: one binding site for BsaI will still be present in the insert after digestion
```

Figure 3 *GG_assembly.m* is directed to control possible mistakes and limit user-control. In contrast to Benchling Assembly Tool, *GG_assembly.m* gives an error if one binding site for BsaI is present. The screenshot (Benchling) is taken from the training video on the reference page.

Also, the Benchling tool lacks an efficient control step for the assessing compatibility of the sticky-ends of different junctions in the one-pot GG assembly (Figure 4). Speculatively, it increases the possibility of an incorrect assembly in experimental conditions.

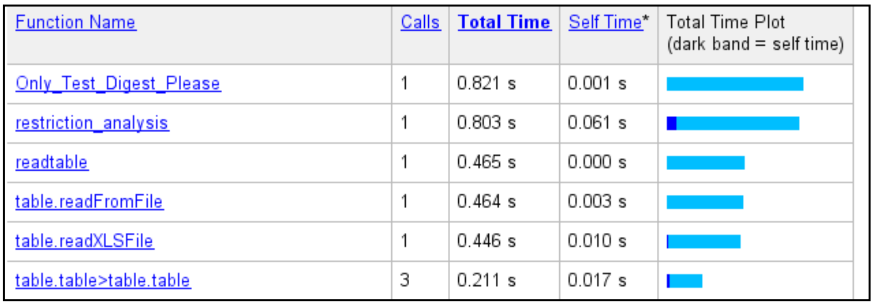


```
>> sticky_array = {'CCTG', 'AACT','GGAA'};
>> highest_thres = 0;
>> [compatability, non_comp_sticky] = check_sticky(sticky_array, highest_thres);
>> compatability, non_comp_sticky
compatability =
     1
non_comp_sticky =
  0×0 empty cell array
>> sticky_array = {'CCTG', 'CCAC','GGAA'};
>> [compatability, non_comp_sticky] = check_sticky(sticky_array, highest_thres)
compatability =
     0
non_comp_sticky =
  1×2 cell array
    {'CCTG'}    {'CCAC'}
>> [compatability, non_comp_sticky] = check_sticky(sticky_array, 1)
compatability =
     1
non_comp_sticky =
  0×0 empty cell array
```

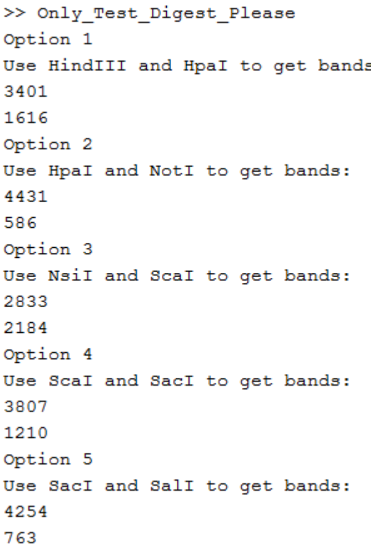
Figure 4 GG_assembly has an additional 'quality control' step in comparison to the Benchling tool. Sticky ends obtained with *GG_assembly.m* (upper code lines: CCTG, AACT, GGAA) can be used in one-pot reaction with the highest fidelity, where is sticky ends derived from Benchling tool (CCTG, CCAC GGAA) passed "check_sticky.m" control only due to increased threshold for allowing more non-specific ligation events (from 0 to 1).

3.2. Created a tool for automatized restriction analysis (*Only_Test_Digest_Please.m*)

The tool "*Only_Test_Digest_Please.m*" takes a sequence and outputs two optimal type IIP enzymes for restriction analysis and bands (size, bp) that are supposed to be seen after the test digest (Figure 5). Besides the model described in the Methods for optimization of the distance between the neighboring bands, it uses a threshold to eliminate poorly visible (due to accumulation of less quantity of the intercalating dye) 'light' bands.



Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
Only_Test_Digest_Please	1	0.821 s	0.001 s	▬
restriction_analysis	1	0.803 s	0.061 s	▬
readtable	1	0.465 s	0.000 s	▬
table.readFromFile	1	0.464 s	0.003 s	▬
table.readXLSFile	1	0.446 s	0.010 s	▬
table.table>table.table	3	0.211 s	0.017 s	▬



```
>> Only_Test_Digest_Please
Option 1
Use HindIII and HpaI to get bands
3401
1616
Option 2
Use HpaI and NotI to get bands:
4431
586
Option 3
Use NsiI and ScaI to get bands:
2833
2184
Option 4
Use ScaI and SacI to get bands:
3807
1210
Option 5
Use SacI and SalI to get bands:
4254
763
```

Figure 5. "Only_Test_Digest_Please.m" allows receiving optimal combinations of type IIP restriction enzymes in less than 1 sec

4. Discussion

Overall, this work provides an efficient, high-fidelity, and automatized tool for the Golden gate assembly. However, it has one major disadvantage in comparison to the Benchling tool [4] – *GG_assembly.m* outputs assembled sequence without keeping annotations of the inputs (vector and inserts). Thus, a user might need to do an assemble with the suggested primers "manually" (in ApE, or Benchling [4,5]), and then align to the sequence from *GG_assembly.m* (in order to check the accuracy of manipulations). Benchling, although outputs somewhat wrong assemblies (please see the section 3.1 in Results), does it with keeping annotations.

In Matlab, functions that structure gb-files (e.g., genbankread) rely on searching for the keywords in the file (such as, LOCUS, ORIGIN) and work poorly with user-annotated files; my attempts to write a genbankread-like function failed due to variety of syntaxis that takes place in the file features (annotations of sequence). The output of a .txt-file (.gb and .ape belong to ASCII files) with initial annotations is incorrect since annotations are applied to the nucleotide order in the input file and not corrected due to all the manipulation events ('digestion,' 'ligation' and others).

References:

1. HamediRad M, Weisberg S, Chao R, Lian J, Zhao H. Highly Efficient Single-Pot Scarless Golden Gate Assembly. ACS Synthetic Biology. 2019;8(5):1047-1054.

2. Potapov V, Ong JL, Langhorst BW, et al. A single-molecule sequencing assay for the comprehensive profiling of T4 DNA ligase fidelity and bias during DNA end-joining. Nucleic acids research. 2018;46(13):e79-e79.

3. Potapov V, Ong JL, Kucera RB, et al. Comprehensive Profiling of Four Base Overhang Ligation Fidelity by T4 DNA Ligase and Application to DNA Assembly. ACS Synthetic Biology. 2018;7(11):2665-2674.

4. Benchling Assembly Wizard (by NEB): benchling.com/tutorials/11/golden-gate-assembly

5. A plasmid Editor (ApE): https://jorgensen.biology.utah.edu/wayned/ape/

Project Code Repository:

https://github.com/AlenaStreletskaia/Final-Project