

Tutorial for Modular Cloning

1 GENERAL INFORMATION

This tutorial is meant for both beginning and advanced users of the Modular Cloning or MoClo system and contains all the necessary information for the *in silico* design, simulation and the practical work involved in Modular Cloning.

For additional information on the MoClo system and other cloning methods, read “Lee et al. 2015: A Highly Characterized Yeast Toolkit for Modular, Multipart Assembly” and “Casini et al. 2015: Bricks and blueprints: methods and standards for DNA assembly”.

2 OVERVIEW OF THE PARTS

2.1 THE MOCLO ENTRY VECTOR DATABASE

In order to assemble your desired expression vector, the entry vectors that harbour the individual parts need to be available. All available parts are currently listed in the BCCM plasmid collection database (<https://bccm.belspo.be/catalogues/plasmid-sets/openpichia>).

2.2 PART TYPES AND PRIMERS

First of all, it is important to allocate a part type to your fragment of interest (i.e. the function of that part in the final expression vector). The following part types are available:

- 1) Part 1 (Left Assembly Connector): This is a connector sequence that allows the user to incorporate several expression cassettes on a single vector, by carefully choosing this part and part 5 (Right Assembly Connector; see “Multigene Plasmid Assembly”). If only one expression cassette is desired, you can simply use the pPTK070 (ConLS) and pPTK074 (ConRE).
- 2) Part 2 (Promoter): Although this part speaks for itself, be aware that the 4 bp overhang between this part and part 3 (or 3a) automatically generates a startcodon (check this yourself as well). The system was designed as such to easily incorporate an N-terminal fusion to your CDS afterwards, without having to remove the startcodon of your CDS (part 3b) first.
- 3) Part 3 (Coding Sequence): Although you can opt for just putting your CDS here (without startcodon, see above), options were included in the system to clone an N-terminal and/or a C-terminal tag to your protein CDS. In the system, an N-terminal tag is denominated as part 3a, while the C-terminal tag is part 4a. The protein CDS is declared a part 3b. If no N-terminal fusion is desired, the CDS is declared as a part 3. Do not include a stopcodon as part 4 will automatically provide one.
- 4) Part 4 (Terminator): Similar to part 3, two subtypes are available here, this time to allow the incorporation of a C-terminal fusion. Therefore, the C-terminal fusion is termed part 4a and

the terminator part 4b. If no C-terminal fusion is used, the terminator is instead declared as part 4. In both setups however, a stopcodon will be automatically introduced (please verify!).

- 5) Part 5 (Right Assembly Connector): See Part 1
- 6) Part 6 (Pichia Selection Marker): This part contains a selectable marker for *Pichia*. Every marker of this type is (and should be) consisting of an appropriate promoter, ORF and terminator. You can use a stuffer part 6 if the *E. coli* marker can also be used for *Pichia* (i.e. if the antibiotic system works and if an eukaryotic promoter is included) – see Part 8.
- 7) Part 7 (Miscellaneous): This is perhaps the most versatile part of the system since it can fulfill multiple roles. For now, options are included to integrate the vector into one of *Pichia*'s chromosomes or to autonomously replicate (ppARS). For most expression vectors however, the stuffer sequence will be the part that you need.
- 8) Part 8 (*E. coli* marker and ORI): This part is meant for propagation of the plasmid in *E. coli*. Note that the Zeocin marker also has a yeast promoter. This way, part 6 can be replaced with a short stuffer which makes the plasmid size considerably smaller. Also 'floxed' versions of selection markers are available.

(More information on part types can be found in the supplementary information of Lee et al. 2015.)

3 HOW TO CREATE AN ENTRY VECTOR

3.1 DESIGN

Once you know the part type of your part of interest, it is time to design appropriate primers. Start with designing standard primers to amplify your template (e.g. gDNA or gBlock). When multiple parts will be generated, it is recommended to adjust primer lengths to obtain a 'consensus' melting temperature, in order to perform all PCRs simultaneously.

For convenience, an depiction of an Excel file to generate PCR primers can be found in the addendum. Use the first column for primer name(s). Search for the part specific overhangs and copy paste the appropriate ones to the second column. Use the CONCATENATE function of Excel in the fourth column to achieve the full length primers. The primers should look similar to the ones shown in Figure 1.

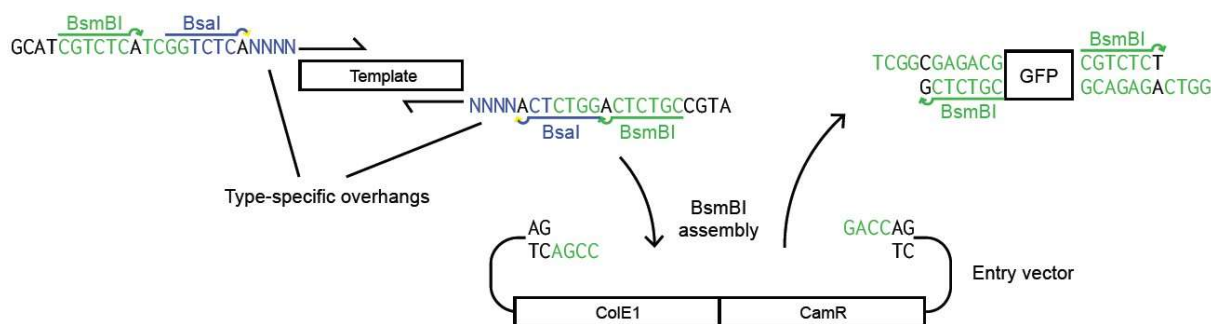


Figure 1. Overview of the BsmBI assembly reaction mechanism (Lee et al. 2015 Supplementary Information).

3.2 *IN SILICO* SIMULATION

It's strongly recommended to generate every PCR product *in silico* first and to import these into the open source **Benchling** platform (Figure 2). The Benchling software allows to simulate the Golden Gate cloning of the PCR product and the recipient entry vector e.g. pPTK081 (which is, in principal, identical to the more 'restricted' pYTK001 from Addgene). For this simulation, open both the PCR product of interest and the recipient entry vector. Start the **Assembly Wizard** at the bottom right corner of the interface. Choose **Create New Assembly** and then **Golden Gate**. Now, in the **Overall Assembly** interface, click on the gearing wheel to open the options and choose the appropriate Type II restriction enzyme (**BsmBI**). In the interface **Set Fragment**, set the recipient vector as the 'backbone' and your PCR product as the 'insert'.

Remember that you first have to open the DNA sequence that needs to be set in order for this to work. Check the **Overall Assembly** interface for any feedback. In case of a compatible assembly the feedback should be 'looks like everything checks out'. You can then click the **Assembly** button, rename the file (default is 'untitled assembly') and save it in a folder. Below is the basic Benchling interface and where to find the tools needed for a Golden Gate cloning.

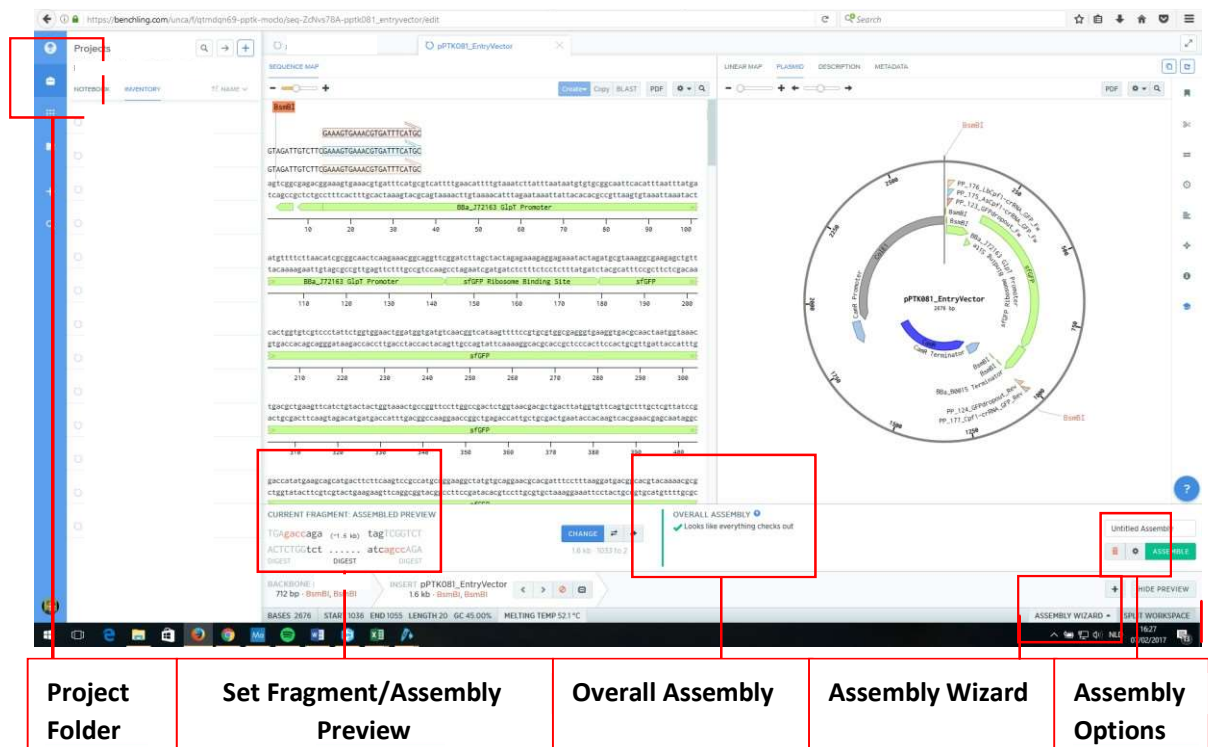


Figure 2. General interface of the open source Benchling platform during a BsmBI assembly. Encircled items are the main tools/interfaces required for a Golden Gate Assembly.

3.3 DESIGN OF OLIGONUCLEOTIDES

Small parts (e.g. N- and C-terminal tags consisting of only tens of bp) can be assembled as well by using the same design as for PCR products. However, as oligos can be custom made, they can be designed as such to fit directly into the recipient entry vector, without the requirement of BsmBI restriction sites. As a consequence, the overhangs can be reduced by 11 nucleotides on both ends reducing overall oligo costs (especially in the 60-90 nucleotide range). The overhang should still contain the right BsaI restriction sites, as well as the four nucleotide overhangs (5'TCGG'3 and 5'GGTC'3) to fit into

the entry vector (Figure 3). As this vector still contains the BsmBI sites, the same BsmBI assembly protocol should be used for both PCR products as oligonucleotides (see next subtitle).

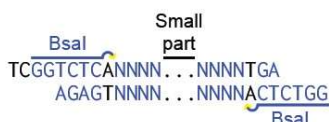


Figure 3. Design of oligonucleotides for assembly into the recipient entry vector e.g. pYTK001/pPTK081 (Lee et al. 2015 Supplementary Information). The overhangs contain two features: the correct Bsal restriction sites (and eventual Bsal generated overhangs) and the four nucleotide overhangs to fit into the recipient entry vector.

Note: The BsmBI and Bsal sites should only be present in the 5' overhangs and should be avoided in the sequence of interest when designing oligos or gBlocks. Additional BsmBI and Bsal restriction sites would increase the possibility of incorrectly assembled entry or expression vectors, respectively. When amplifying a sequence from a source/template, the BsmBI/Bsal sites cannot be removed in the design of course. Instead, you can remove them by overlap extension PCR (OE PCR). In OE PCR, the sequence is divided in two fragments, approximately at the site of the undesired Type II restriction site. Both these fragments should be amplified with primers that 1) generate a >30 bp overlap region between the two fragments and 2) eliminate the restriction site by incorporating a (silent) mutation in this overlap region. Perform a second PCR by pooling both fragments and use a low annealing temperature and long annealing time (e.g. 52°C for 45 sec annealing).

3.4 BSMBI ASSEMBLY PROTOCOL

Use the designed primers to generate your PCR product. Use a total PCR volume of 50 µL and analyze 10 µL on gel. Purify the remaining 40 µL, using the MagBio® HighPrep™ PCR protocol (magnetic beads). Use the purified PCR product in the reaction mix mentioned below (Figure 4) and transform *E. coli* with the assembled vector. Formula for x ng = (650/1000) * length (bp).

Note: In case the part of interest is generated by annealing two oligonucleotides, simply use 2µL of each primer in 18 µL of duplex buffer. Anneal in the PCR cyclor (95°C ramp down to 4°C over 20 min). Then use >1000 femtomole of the oligos in the assembly reaction.

Modular Pichia parts system - BsmBI assembly			
Date:			
	Reaction Mix	Reaction program	
BsmBI (NEB)	1 µl	1. 42°C for 2 minutes	Digestion
T4 DNA ligase (NEB)	1 µl	2. 16°C for 5 minutes	Ligation
10x T4 DNA ligase buffer (NEB)	2 µl	3. Go to 1 for minimum 25 times	
PCR insert	> 1000 fmol	4. 60°C for 10 minutes	Final digestion
pYTK001 (20 fmol/ul)	1 µl	5. 80°C for 10 minutes	Heat denaturation
	Add H2O until 20 µl	6. 12°C forever	
For high efficiencies, use much more of the insert compared to the vector backbone (pYTK001) (as much as possible) !			
Transform (heat shock) assembly mix to competent <i>E. coli</i> MC1061 (!!) cells.			
NOTE: MC1061 seem to work OK; DH5a cells were tried as well, but green-white screening was quite difficult			
Plate 20 and 100 µl of the final 1 ml culture mix on LB plates supplemented with the appropriate antibiotic			
Pick white (non-green) colonies for inoculation, plasmid prep and sequencing (primers PP_001 and PP_002).			
OPTIONAL: control digest (with Bsal sites)			

Figure 4. BsmBI assembly protocol to create an entry vector.

In theory, the subsequent cycles of digestion and ligation will enrich for correctly assembled vectors. This is due to absence of the recognition site of the Type II restriction enzyme in the assembled vector, while it is still present in the undigested vector. In case of a (successful) assembly, the GFP cassette of

the e.g. pYTK001/pPTK081 vector should be replaced with your fragment of interest. Therefore, grow the *E. coli* cells overnight on selective medium (LB + antibiotic) and screen for white colonies. In our experience, MC1061 *E. coli* cells work fine, while efforts with the DH5 α were significantly less successful. Inoculate 3 clones of each construct. Inoculating more is usually not necessary as the clone to clone variation appears to be rather low.

Note: For small colonies, the green-white screening can be somewhat difficult. In our experience, storing the plates at 4°C for several hours increases the contrast between green and white colonies.

Sequence every generated entry using primers PP001 (5'GAGCCTATGGAAAAACGCCAGC'3) and PP002 (5'GATATCGAGCTCGCTTGGACTCC'3) and save the sequencing data on your computer.

4 HOW TO CREATE AN EXPRESSION VECTOR

4.1 DESIGN AND *IN SILICO* SIMULATION

Once all required entry vectors are generated, it's time to simulate the assembly of your expression vector in Benchling (Figure 5). Open all entry vectors of interest so that each part type (or number in the final assembly) is represented. There should be 8 to 10 entry vectors opened in the Benchling interface, depending on the presence of any N- and/or C-terminal fusions.

Again, start the **Assembly Wizard** at the bottom right corner of the interface. Choose **Create New Assembly** and then **Golden Gate**. Now, in the **Overall Assembly** interface, click on the gearing wheel in the bottom right corner to open the options and choose the appropriate Type II restriction enzyme (**BsaI**). In the interface **Set Fragment**, set the 'part 8' entry vector (*E. coli* marker + ORI) as the '*backbone*' and all other entry vectors as '*inserts*'. You can add multiple insert by clicking the 'plus' sign in the bottom right corner. To set a fragment as insert, the corresponding file of the entry vector needs to be the one you're viewing at that moment. Note that the program is somewhat blind for the order of all parts in the final assembly so you have to add the '*inserts*' in the right order yourself (e.g. part 8 -> part 1 -> ... -> part 7). In order for the parts to fit altogether, you will often have to reverse the part (some parts are encoded by the + strand; other by the - strand). To reverse the part, set the part as an insert and then click the two arrows button in the **Set Fragment** interface.

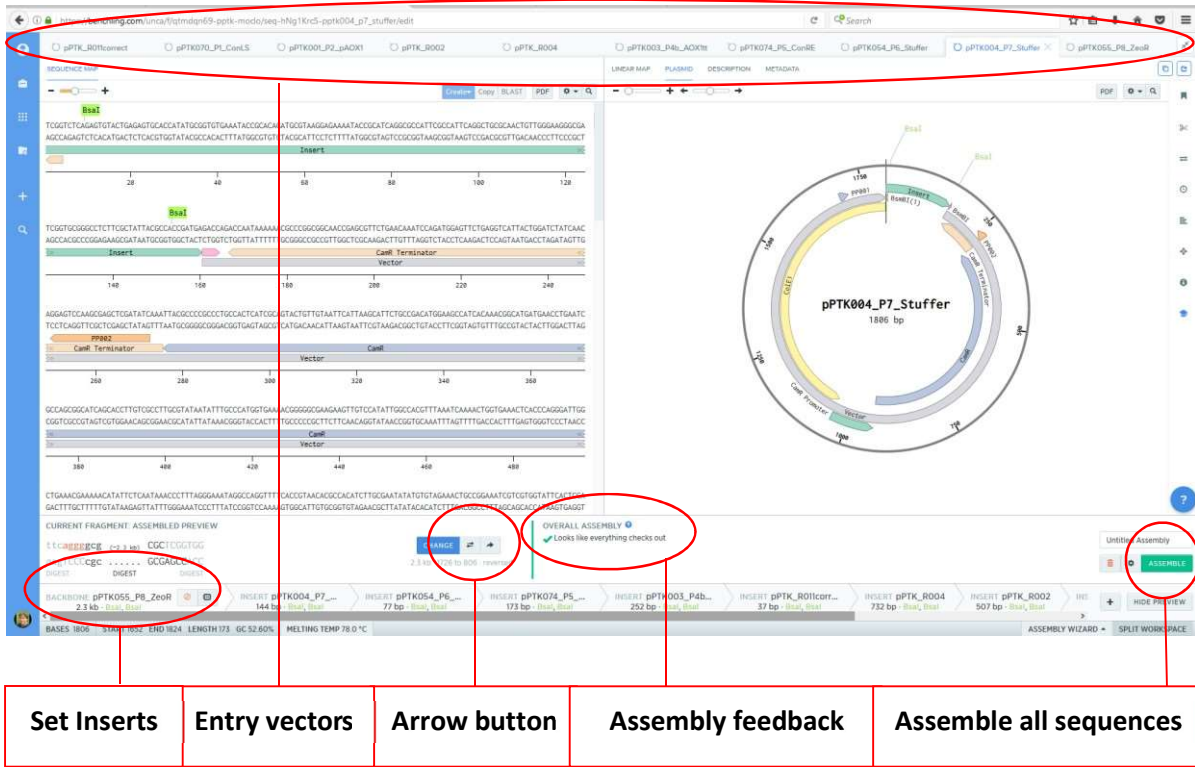


Figure 5. General interface of the open source Benchling platform for a BsaI assembly. Encircled items are tools and points of interest that are important during an assembly.

Now check your (theoretical) expression vector for the correct orientation of the parts (especially the expression cassette) and look for good restriction sites (e.g. a RE that is unique for your part of interest or just gives the most information). As all entry vectors were sequenced and no further PCR-based cloning steps were used, sequencing the expression vector should be unnecessary. The assembly protocol is similar to the BsmBI assembly (Figure 6).

Modular Pichia parts system - BsaI assembly

	Reaction Mix	Reaction program	
BsaI (NEB)	1 μ l	1. 37°C for 2 minutes	Digestion
T4 DNA ligase (NEB)	1 μ l	2. 16°C for 5 minutes	Ligation
10x T4 DNA ligase buffer (NEB)	2 μ l	3. Go to 1 for minimum 25 times	
Parts (20 fmol/μl)	1 μ l/part	4. 60°C for 10 minutes	Final digestion
	Add H₂O until 20 μl	5. 80°C for 10 minutes	Heat denaturation
		6. 12° forever	

Transform (heat shock) assembly mix to competent E. coli MC1061 (!!) cells.

NOTE: MC1061 seem to work OK; DH5a cells were tried as well, but red-white screening was quite difficult

Pick white (non-red) colonies for inoculation and plasmid prep.

QUALITY CONTROL: sanger sequencing of expression construct (optional) and control digest (e.g. using unique cutter and BsmBI (cuts twice and only twice, once in each assembly connector (part 1 and 5))

Note: Do not use more of the part 8 or even use only 0.5 μ l instead of 1 μ l to reduce the number of false (red) colonies

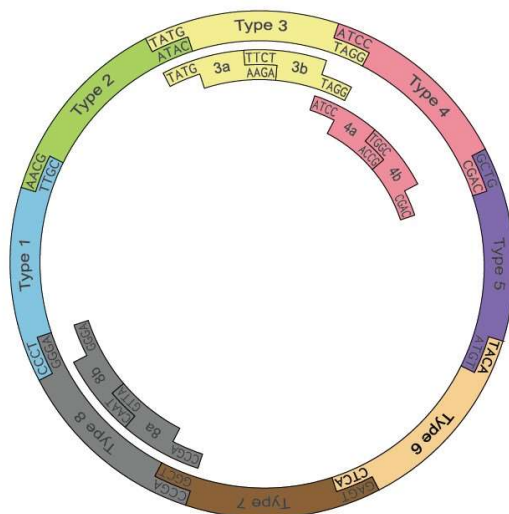
Figure 6. BsaI assembly protocol to create an expression vector.

5 ADDENDUM

5.1 CONSTRUCTION OF PARTS – PRIMER FILE

What?	Primer name	5' Overhang	4 bp part-specific overhang	Template binding region	Full-length primer
Left Assembly connector	P1_L	GCATCGTCTCATCGGTCTCA	CCCT	Nx	GCATCGTCTCATCGGTCTCACCTNx
	P1_R	ATGCCGTCTCAGGTCTCA	CGTT	Nx	ATGCCGTCTCAGGTCTCACGTTNx
Promoter	P2_L	GCATCGTCTCATCGGTCTCA	AACG	Nx	GCATCGTCTCATCGGTCTCAAACGNx
	P2_R	ATGCCGTCTCAGGTCTCA	CATAgatct	Nx	ATGCCGTCTCAGGTCTCACATAgatctNx
GOI	P3_L	GCATCGTCTCATCGGTCTCA	TATG	Nx	GCATCGTCTCATCGGTCTCATATGNx
	P3_R	ATGCCGTCTCAGGTCTCA	GGATcc	Nx	ATGCCGTCTCAGGTCTCAGGATccNx
N-terminal fusion	P3a_L	GCATCGTCTCATCGGTCTCA	TATG	Nx	GCATCGTCTCATCGGTCTCATATGNx
	P3a_R	ATGCCGTCTCAGGTCTCA	AGAAcc	Nx	ATGCCGTCTCAGGTCTCAAGAAccNx
GOI when N-terminal fusion	P3b_L	GCATCGTCTCATCGGTCTCA	TTCT	Nx	GCATCGTCTCATCGGTCTCATTTNx
	P3b_R	ATGCCGTCTCAGGTCTCA	GGATcc	Nx	ATGCCGTCTCAGGTCTCAGGATccNx
Terminator	P4_L	GCATCGTCTCATCGGTCTCA	ATCCtaa	Nx	GCATCGTCTCATCGGTCTCAATCCtaaNx
	P4_R	ATGCCGTCTCAGGTCTCA	CAGC	Nx	ATGCCGTCTCAGGTCTCACAGCNx
C-terminal fusion	P4a_L	GCATCGTCTCATCGGTCTCA	ATCC	Nx	GCATCGTCTCATCGGTCTCAATCCNx
	P4a_R	ATGCCGTCTCAGGTCTCA	GCCAtta	Nx	ATGCCGTCTCAGGTCTCAGCCAttaNx
Terminator	P4b_L	GCATCGTCTCATCGGTCTCA	TGGC	Nx	GCATCGTCTCATCGGTCTCATGGCNx
	P4b_R	ATGCCGTCTCAGGTCTCA	CAGC	Nx	ATGCCGTCTCAGGTCTCACAGCNx
Right Assembly connector	P5_L	GCATCGTCTCATCGGTCTCA	GCTG	Nx	GCATCGTCTCATCGGTCTCAGCTGNx
	P5_R	ATGCCGTCTCAGGTCTCA	TGTA	Nx	ATGCCGTCTCAGGTCTCATGTANx
Pichia selection marker	P6_L	GCATCGTCTCATCGGTCTCA	TACA	Nx	GCATCGTCTCATCGGTCTCATACANx
	P6_R	ATGCCGTCTCAGGTCTCA	ACTC	Nx	ATGCCGTCTCAGGTCTCAACTCNx
Miscellaneous	P7_L	GCATCGTCTCATCGGTCTCA	GAGT	Nx	GCATCGTCTCATCGGTCTCAGAGTNx
	P7_R	ATGCCGTCTCAGGTCTCA	TCGG	Nx	ATGCCGTCTCAGGTCTCATCGGNx
E. coli marker and origin	P8_L	GCATCGTCTCATCGGTCTCA	CCGA	Nx	GCATCGTCTCATCGGTCTCACCGANx
	P8_R	ATGCCGTCTCAGGTCTCA	AGGG	Nx	ATGCCGTCTCAGGTCTCAAGGGNx
	P8a_L	GCATCGTCTCATCGGTCTCA	CCGA	Nx	GCATCGTCTCATCGGTCTCACCGANx
	P8a_R	ATGCCGTCTCAGGTCTCA	ATTG	Nx	ATGCCGTCTCAGGTCTCAATTGNx
	P8b_L	GCATCGTCTCATCGGTCTCA	CAAT	Nx	GCATCGTCTCATCGGTCTCACAAATNx
	P8b_R	ATGCCGTCTCAGGTCTCA	AGGG	Nx	ATGCCGTCTCAGGTCTCAAGGGNx

5.2 VISUAL REPRESENTATION OF AN ASSEMBLED MOcLO EXPRESSION VECTOR



5.3 FLUORESCENT SCREENING OF ASSEMBLED VECTORS

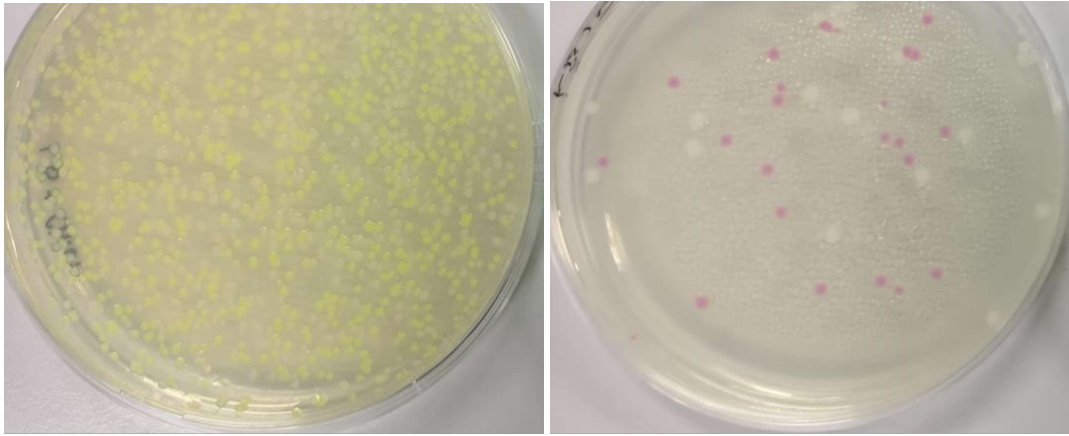


Figure 7. Fluorescent Screening of assembled vectors. The left figure depicts the green-white screening of a BsmBI assembly. The right figure shows a red-white screening, following a BsaI assembly.