Chapter 2

Combinatorial-Hierarchical DNA Library Design Using the TeselaGen DESIGN Module with *j5*

Michael J. Fero, James K. Craft, Trang Vu, and Nathan J. Hillson

Abstract

Modern DNA assembly techniques are known for their potential to link multiple large DNA fragments together into even larger constructs in single pot reactions that are easier to automate and work more reliably than traditional cloning methods. The simplicity of the chemistry is in contrast to the increased work needed to design optimal reactions that maximize DNA fragment reuse, minimize cost, and organize thousands of potential chemical reactions. Here we examine available DNA assembly methods and describe through example, the construction of a complex but not atypical combinatorial and hierarchical library using protocols that are generated automatically with the assistance of modern synthetic biology software.

Key words DNA assembly, Pathway engineering, Synthetic biology, Metabolic engineering, Bioinformatics, Cloning

1 Introduction

It is common in the parlance of synthetic biology to speak in terms of forward engineering biological systems in much the same way that an electrical engineer might design and build an electronic circuit [1]. Progress has been made in translating engineering design principles to the description of biological parts and devices as well as systems [2]. Nevertheless, a biological system is not as easily characterized or simulated as an electronic or mechanical system. Progress often involves building many variants of a genetic construct or pathway, and then performing functional tests in order to find the DNA sequence that encodes the best performing biological system. DNA synthesis costs are dropping rapidly, but direct synthesis alone cannot address the combinatorial explosion that occurs when trying to optimize a biological pathway that may contain even a small number of variants tested in combinatorial fashion, leading to libraries of thousands of elements. The cost of building a large library grows linearly with library size without the benefit of part reuse. Streamlined part reuse means that library costs

grow only logarithmically with increasing library size, enabling large screens at low cost. Importantly, even the everyday construction of small libraries or single constructs is enabled more effectively and without error by using modern DNA assembly methods and software [3].

A common challenge is to (a) design a set of DNA constructs that span the space of potentially optimal constructs for the performance of a particular function, (b) build the constructs minimizing errors, cost and time, (c) transform into a host chassis (*E. coli*, yeast), (d) perform a functional assay to determine individual construct efficacy, (e) capture the results in a DNA parts knowledgebase, and (f) use those learnings to inform further rounds of optimization or investigation. The general problem of assembling a set of somewhat arbitrary DNA "parts" into a contiguous construct is shown in Fig. 1. Two key observations inform our response to this challenge.

1. DNA synthesis costs will track DNA sequencing costs and will continue to drop while the length of reliably synthesized constructs will increase. DNA synthesis improvements, coupled with the combinatorial DNA assembly techniques described here, will help to make building complex DNA libraries faster and easier.
2. Our knowledge of biology is incomplete. Accurate simulations of biological systems, while improving, will lag behind our ability to quickly build and test, implying that new approaches

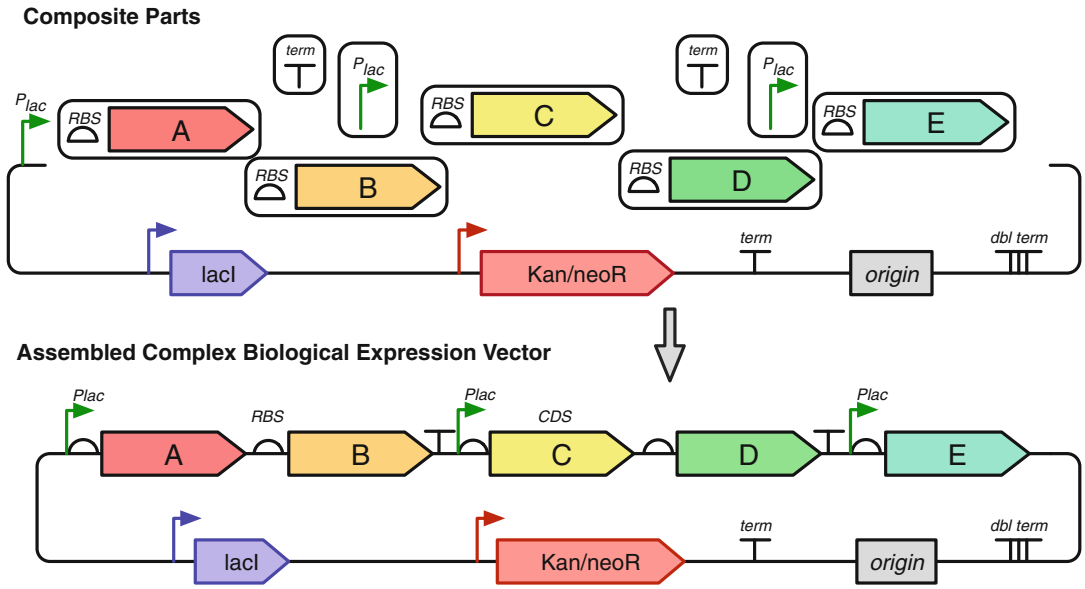

Fig. 1 Multipart DNA assembly

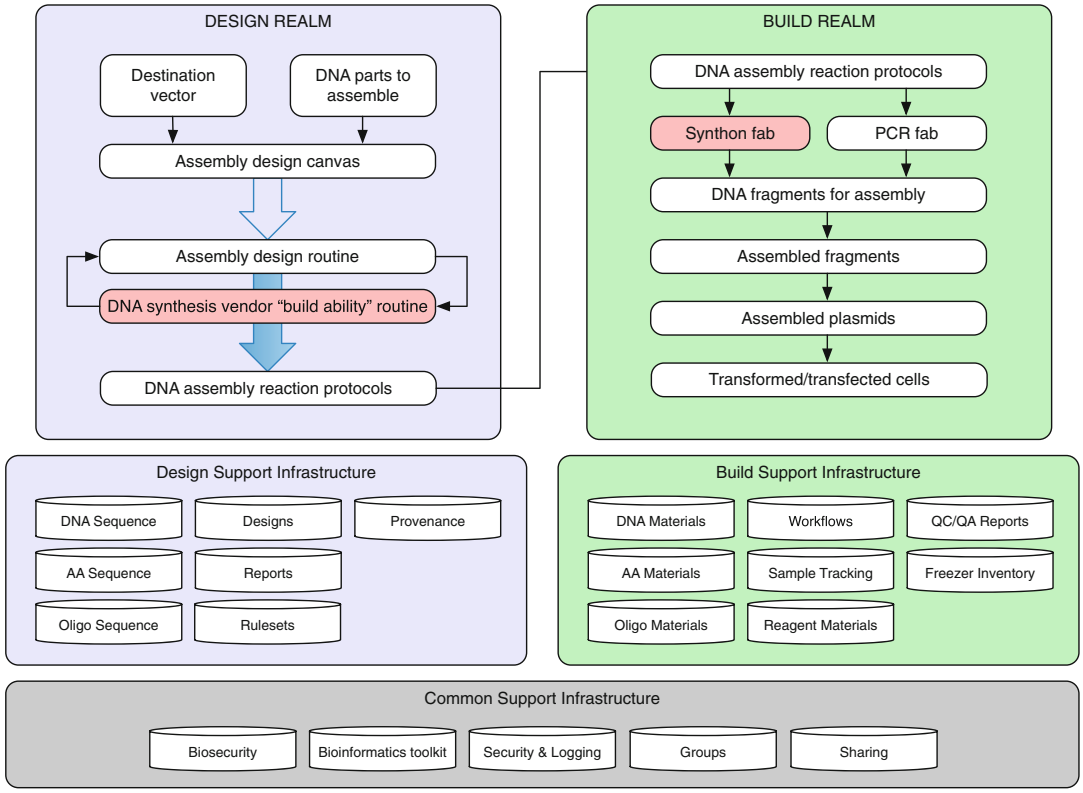

Fig. 2 Design/Build information technology elements

to engineering biology via intelligent, iterative, high content screens will outpace raw simulation.

These considerations imply that a modular but tight coupling between DNA design, cell construction, testing, and learning will allow researchers to converge quickly on a bioengineered product of interest. Modern synthetic biology software can help organize and inform this process. A schematic including common elements of a platform that can organize the bioCAD/CAM or Design/Build process is shown in Fig. 2.

2 DNA Design and Build Methods

2.1 Traditional Multiple Cloning Site Approach to DNA Assembly

In order to build libraries in ways that facilitate larger scales and automation, we avoid designing and building DNA constructs using the traditional MCS (Multiple Cloning Site) approach. To understand why, we briefly review the traditional approach. In a typical cloning vector, the MCS follows a promoter (e.g., a T7 promoter) and is in turn followed by a terminator. Integrating a protein coding sequence of interest into a plasmid vector involves:

(1) identification of two restriction sites present in the MCS, but absent in the coding sequence of interest, (2) PCR amplification of the coding sequence of interest with DNA oligo primers flanked with the selected restriction sites, (3) digestion of the PCR product as well as the destination vector with the *corresponding* restriction enzymes, and (4) ligation of the purified digested PCR product and destination vector. This approach works well for integrating a single coding sequence into the MCS of the destination expression vector but becomes an increasingly poor choice with each additional sequence fragment to be assembled. Integrating 10 fragments into a plasmid vector, for example, would require 11 restriction sites with distinct overhang sequences, including two from the MCS, with the additional requirement that each is absent from flanking assembly fragments. Also, in traditional cloning every different assembly might require a different combination of restriction enzymes, reaction temperatures, and buffer conditions. In general, it is unlikely that a single enzymatic mix can be applied across independent assemblies, making the process less amenable to parallelization and automation. In contrast, the newer methods employed below use a standardized set of enzymes and reaction conditions for every assembly facilitating sequence reuse, parallelization, and automation.

2.2 Flanking Homology Methods (SLIC, Gibson, CPEC, SLiCE; GeneArt, In-Fusion)

SLIC, Gibson, CPEC, and SLiCE (and GeneArt Seamless and In-Fusion) are related methods that offer standardized, scarless, sequence independent, multipart DNA assembly.

2.2.1 SLIC

SLIC, or sequence and ligase independent cloning [4], as its name implies, does not utilize restriction enzymes or ligase. A DNA sequence fragment to be cloned into a destination vector is PCR amplified with oligos whose 5′ termini contain about 25 bp of sequence homology to the ends of the destination vector, linearized either by restriction digest or PCR amplification as shown in Fig. 3. The linearized destination vector and the PCR product containing part A are separately treated with T4 DNA polymerase in the absence of dNTPs. In the absence of dNTPs, T4 DNA polymerase has 3′ exonuclease activity, which begins to chew back the linearized destination vector and the PCR product from 3′ to 5′. Once the termini of the linearized destination vector and the PCR product have sufficient complementary single-stranded 5′ overhangs exposed, dCTP is added to arrest the chew-back reaction. With the addition of dCTP, the T4 DNA polymerase changes activity from 3′ exonuclease to polymerase but stalls because not all dNTPs are present, retaining most, if not the entirety, of each chewed-back overhang. Alternatives to the 3′ chew-back with T4 DNA

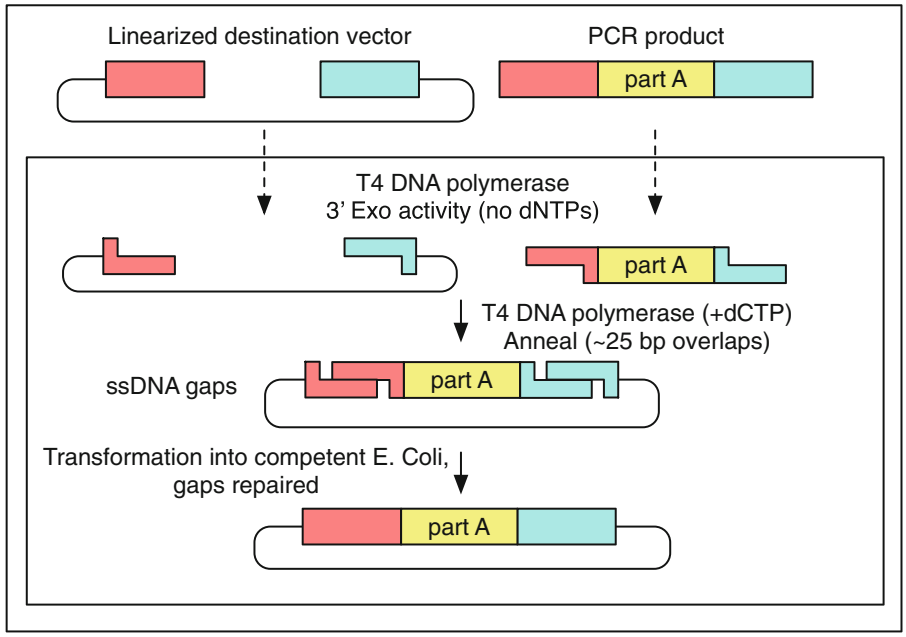

Fig. 3 SLIC assembly of part A with a linearized destination vector

polymerase in the absence of dNTPs include the use of mixed or incomplete PCR products (so this does not apply to the linearized vector backbone if it is derived from a restriction enzyme digest), which can also result in the desired 5′ overhangs, as described in the original SLIC publication [4]. The chewed-back linearized destination vector and PCR product are mixed together and annealed to each other. Since there is no ligase in the reaction, this results in a plasmid with four single-stranded gaps or nicks. Once transformed into competent *E. coli*, the gaps are repaired. Note that SLIC assembly is standardized, in that it always uses the same reaction components and conditions, scarless, since there is no sequence in the resulting assembly that is not user-designed, and sequence-independent, as the method is not (at least to a large extent, but see below) sensitive to the sequences of either the destination vector or the part to be incorporated.

2.2.2 *Gibson*

Gibson DNA assembly, named after the developer of the method [5] is analogous to SLIC, except that it uses a dedicated exonuclease (no dNTP addition step), and uses a ligase to seal the single-stranded nicks as shown in Fig. 4. The linearized destination vector and the PCR product containing part A are mixed together with T5 exonuclease, which chews back the linearized destination vector and the PCR product from 5′ to 3′. Phusion polymerase, which (with the annealed linearized destination vector and PCR product effectively priming each other) fills in the gaps, and ligase seals the

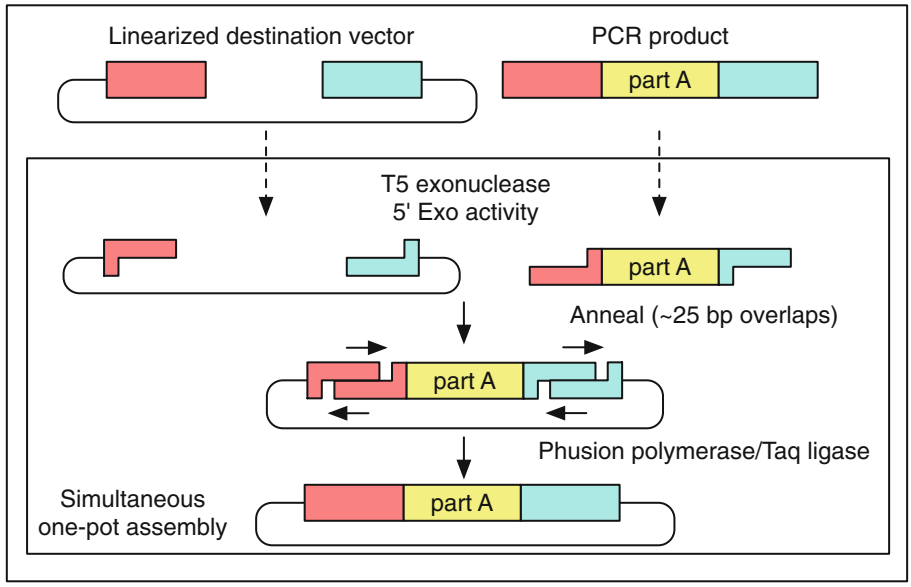

Fig. 4 Gibson assembly of part A with a linearized destination vector

four single-stranded nicks. The polymerase chases the exonuclease around the plasmid, with the polymerase eventually overtaking, as the exonuclease is gradually heat-inactivated (and Phusion is extremely fast). Like SLIC, Gibson assembly is standardized, scarless, and largely sequence-independent. Gibson is advantageous over SLIC in that it is a simultaneous one pot reaction (the two-step addition of dCTP is not required), the presence of ligase may boost assembly efficiency, and since the assembly reaction occurs at an elevated temperature relative to SLIC, there may be fewer problems when somewhat stable secondary structures occur at the ends of assembly pieces. The disadvantage of the Gibson method is that the T5 exonuclease, Phusion polymerase, and Taq ligase cocktail is more expensive than that required for SLIC. An anecdotal/empirical limitation of the Gibson method is that it works best to assemble DNA fragments that are at least 250 bp in length or longer; this is perhaps due to the likelihood that the T5 exonuclease would entirely chew through a short DNA fragment before it has a chance to anneal and prime the Phusion polymerase for extension. While the same could be said for SLIC, the timing of dCTP addition provides some control in switching from the exonuclease to the polymerase activity of T4 DNA polymerase (the use of mixed or incomplete PCR products can prevent this problem all together), although caution should be applied when using SLIC to assemble small DNA fragments. Prior to Gibson (or SLIC) assembly, it is recommended to SOE (splice by overlap extension) together neighboring assembly fragments until their cumulative size is larger than 250 bp. Fortunately, the very same PCR products

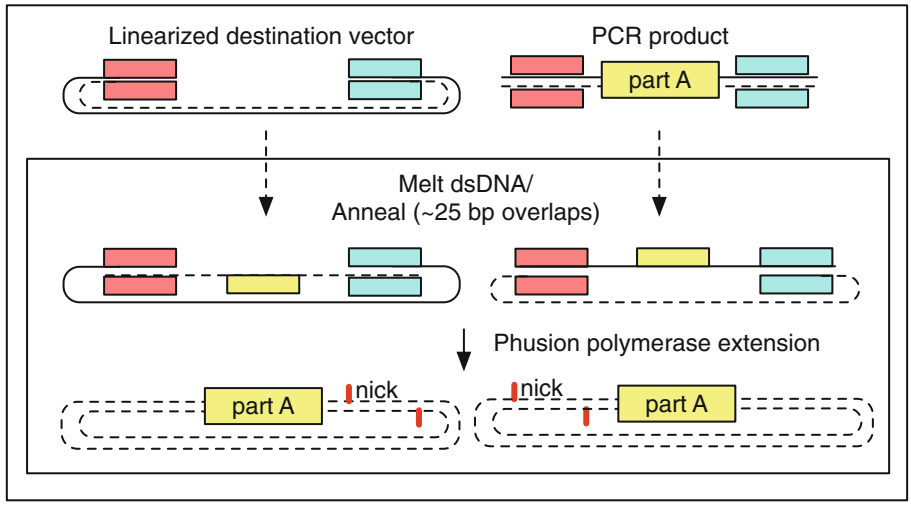

Fig. 5 CPEC assembly of part A with a linearized destination vector

designed for Gibson (and SLIC) assembly, already contain the flanking homology sequences required for SOEing. Gibson has been shown to be a good method for assembling large numbers of dsDNA fragments at once. With the help of whole-genome thermodynamic analysis software PICKY [6] up to 45 fragments have been assembled at once [7]. Modifications to the Gibson protocol have also been useful for assembling large DNA fragments with high GC content [8].

2.2.3 CPEC

CPEC, or circular polymerase extension cloning, shown in Fig. 5, is analogous to SOEing together the fragments to be assembled, except that no oligos are utilized (the linearized destination vector and PCR product prime each other, as in SLIC/Gibson assembly) and there are typically only a few thermocycles required [9]. Since there are no (or very few) reamplifications of a given template sequence, PCR-derived mutations are not propagated as much as one would anticipate for standard SOEing reactions. Like SLIC and Gibson assembly, CPEC is standardized, scarless, and largely sequence-independent. CPEC is advantageous in that, since there is no exonuclease chew-back, small sequence fragments can be assembled directly without a preliminary SOEing step, there is no dNTP addition step (unlike SLIC), there is only a single enzyme (polymerase) required (unlike Gibson), and since the CPEC assembly reaction occurs at higher temperatures than either SLIC or Gibson, stable secondary structures at the ends of assembly pieces are relatively less of a concern. The main disadvantages of CPEC is that it is more likely to result in polymerase-derived mutations than SLIC or Gibson, and mispriming events are now possible anywhere along the sequences of the fragments to be assembled (as opposed

to only at the termini of the fragments), although the Gibson method, depending on how much of a head start the T5 exonuclease has, could suffer from similar drawbacks.

2.2.4 SLiCE

SLiCE [10] uses the same types of DNA starting materials as those used for SLIC, Gibson, and CPEC, and results in the same final product. Unlike SLIC, Gibson, or CPEC, however, SLiCE utilizes bacterial cell extract (i.e., an ex vivo DNA assembly method) making it potentially very cost effective since laboratory bacterial strains can be used as sources for the SLiCE extract. A variation of the SLiCE method, which utilizes PPY, a strain of *E. coli* DH10B that expresses a lambda-red recombination system, as the source of the extract, has been demonstrated to increase the efficiency of SLiCE.

2.2.5 GeneArt® Seamless Cloning

GeneArt® Seamless Cloning is a proprietary assembly methodology developed by Thermo Fisher. This assembly method uses the same types of DNA starting materials as those used for SLIC/Gibson/CPEC/SLiCE and results in the same final product. One key difference is that the recommended overlap length is only 15 bps (enabled by a room temperature assembly reaction), which may prove advantageous over SLIC/Gibson/CPEC/SLiCE from the standpoint of requiring shorter/cheaper DNA oligos and enabling combinatorial assembly designs with sequence diversity close to the ends of the sequence fragments to be assembled. On the other hand, a shorter overlap length may reduce assembly specificity, and depending on the assembly mechanism (proprietary), high self-complementarity or strong single-stranded DNA secondary structure in the overlap region may prove more problematic than for SLIC/Gibson/CPEC/SLiCE. Since the overlap length is shorter (~15 bps) than that generally recommended for SLIC/Gibson/CPEC/SLiCE (~25 bps), applying the SLIC/Gibson/CPEC/SLiCE methods to DNA fragments optimized for GeneArt® Seamless Cloning may not be successful. It is possible to use TeselaGen *j5* design parameters optimized for GeneArt® Seamless Cloning. For more information, see the GeneArt® Seamless Cloning documentation on the Thermo Fisher website.

2.2.6 In-Fusion® Cloning

In-Fusion® Cloning is a proprietary assembly methodology developed by Takara-Clontech. This assembly method uses the same types of DNA starting materials as those used for SLIC/Gibson/CPEC/SLiCE and results in the same final product. One key difference is that the recommended overlap length is only 15 bps (like GeneArt® Seamless Cloning, described above, except it operates at 50 °C like Gibson), which may prove advantageous over SLIC/Gibson/CPEC/SLiCE from the standpoint of requiring shorter/cheaper DNA oligos and enabling combinatorial assembly

designs with sequence diversity close to the ends of the sequence fragments to be assembled. On the other hand, a shorter overlap length may reduce assembly specificity, and depending on the assembly mechanism (proprietary), high self-complementarity or strong single-stranded DNA secondary structure in the overlap region may prove more problematic than for SLIC/Gibson/CPEC/SLiCE. Since the overlap length is shorter (~15 bps) than that generally recommended for SLIC/Gibson/CPEC/SLiCE (~25 bps), applying the SLIC/Gibson/CPEC/SLiCE methods to DNA fragments optimized for In-Fusion® Cloning may not be successful. It is possible to use TeselaGen *j5* design parameters optimized for In-Fusion® Cloning. For more information, see the In-Fusion® Cloning User Manual and design tool, available from the Takara-Clontech website.

2.2.7 Flanking Homology Method Similarities

Despite differences in implementation, flanking homology methods all start with the same starting materials and result in the same final products. Thus, an assembly designed for CPEC will be equally applicable to SLIC or Gibson assembly. In certain situations, combinatorial SLIC/Gibson/CPEC assembly can be a very reasonable and effective choice (see, e.g., [11] for Gibson and [12] for CPEC), if the sequence identity throughout all combinations and assembly junctions is extensive enough not to be a limitation.

2.2.8 Flanking Homology Method Limitations

A major limitation to flanking homology methods is that the termini of the DNA sequence fragments to be assembled should not have stable single-stranded DNA secondary structure, such as a hairpin or a stem loop (as might be anticipated to occur within a terminator sequence), as this would directly compete with the required single-stranded annealing/priming of neighboring assembly fragments. It may be possible to mitigate this by padding these problematic termini with sequence from their neighboring assembly fragments. Repeated sequences (such as the repeated terminators and promoters in the example above) are often obstacles to SLIC/Gibson/CPEC/SLiCE assembly, since assembly is directed by sequence homology, and if two distinct assembly fragments are identical at one terminus this can lead to assemblies that do not contain all of the desired parts or may contain parts arranged in the wrong order (*see* **Note 1**). To circumvent these obstacles, which TeselaGen *j5* refers to as "assembly fragment incompatibilities," it is often necessary to perform a sequential hierarchical assembly so as not to place assembly fragments with identical termini in the same assembly reaction at the same time. When possible, it is better to substitute repeated sequences with sequence pairs that are not identical yet encode comparable biological function; this provides a benefit not only to the DNA assembly process but will also enhance the DNA stability of the resulting construct. Finally, flanking

homology methods might not be the optimal choice for combinatorial assembly if sequence diversity occurs at the very ends of the sequence fragments to be assembled (within about 15 bps of the termini), since this will preclude the reuse of the same homology sequences throughout all of the combinations (*see* **Note 2**). However, in certain situations, combinatorial SLIC/Gibson/CPEC/SLiCE assembly can be a very reasonable and effective choice (*see* [11] for Gibson and [12] for CPEC), if the sequence identity throughout all combinations and assembly junctions is extensive enough not to be a limitation. These limitations, which imply that the SLIC/Gibson/CPEC/SLiCE assembly methods are not completely sequence-independent, are largely addressed by the Golden Gate assembly method.

2.3 Type IIs Endonuclease Methods (Golden Gate, MoClo, GoldenBraid)

2.3.1 Golden Gate

The Golden Gate method [13–15] offers standardized, scarless, multipart DNA assembly and is an good choice for combinatorial library construction. The Golden Gate method relies upon the use of a single type IIs endonuclease, whose recognition sites are distal from their cut sites. The example shown in Fig. 6 uses BsaI, where the recognition sequence "GGTCTC" is separated from its 4-bp overhang by a single bp, and its activity is independent of the sequences of the single bp spacer and the 4-bp overhang. The recognition site for BsaI is not palindromic and is therefore directional. In the notation used here, the recognition site is abstractly represented by a clear rectangle below the dsDNA line and the 4-bp overhang sequence is represented by a shaded box (with different

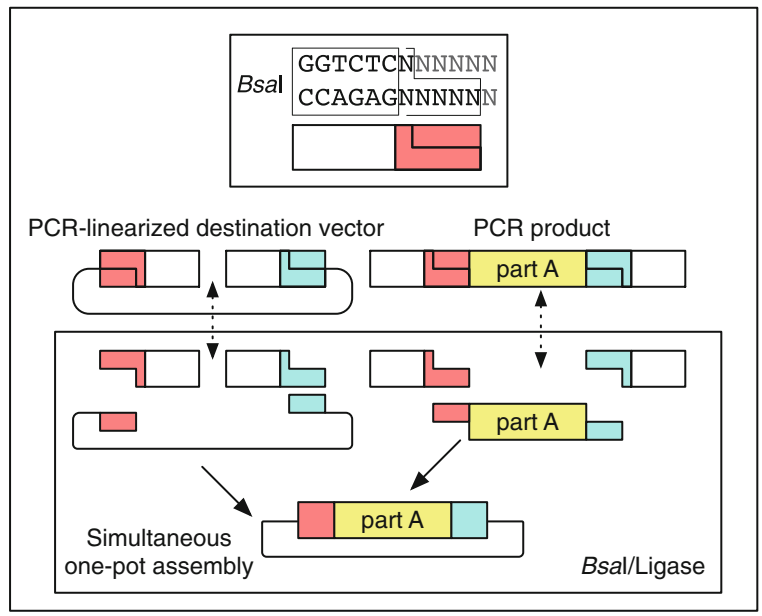

Fig. 6 Golden Gate assembly of part A with a linearized destination vector

shadings indicating different 4-bp sequences). Using this notation, the PCR product containing part A in the example is flanked by two BsaI recognition sites, both pointing inward toward part A. The linearized destination vector is similarly depicted. If the PCR product shown above is mixed with BsaI and ligase, the PCR product is (reversibly) digested, resulting in three DNA fragments, and ligated back together again. The same is true of the linearized destination vector. However, if the PCR product and the linearized destination vector (each of which contains two different 4-bp overhangs) are both mixed together with BsaI and ligase, the cut linearized destination vector will irreversibly ligate (dead-end reaction product) with the cut PCR product containing part A. This particular ligation is irreversible, because the ligation product no longer contains any BsaI recognition sequences. Thus, over time, all reactions will tend toward the desired assembly product. It should be pointed out that the sequences of the 4-bp overhangs are entirely user-specifiable. In this regard, Golden Gate assembly is scarless, since we have complete control over the sequence of the resulting assembly product.

As is true of the flanking homology methods, we can put together many parts at the same time in the same pot (multipart assembly reaction). Golden Gate assembly provides immediate access to every part to be assembled, and with only one transformation step, combinatorial diversity is achieved. Golden Gate assembly is a particularly good choice for constructing combinatorial libraries. Every part in each combinatorial bin is flanked by the same two 4-bp overhang sequences. Any two parts in a bin are completely interchangeable with respect to Golden Gate assembly, and only a single pair of oligos is required for each part across the entire assembly.

Returning to the previous DNA assembly challenge shown in Fig. 1 we can now see that many fragments can be assembled together using Golden Gate, as shown in Fig. 7. In this example, each 4-bp overhang is color-coded, (the BsaI recognition sites, while present and inwardly facing in all of the sequence fragments to be assembled, are not depicted here). We must design the 4-bp overhang sequences for each assembly junction and incorporate them into the 5′ flanking sequence of each oligo, a process that was laborious and error-prone before the advent of assembly software. Note that Fig. 7 is schematic. The assembly junctions must be between DNA fragments that are assembled in the assembly reaction but do not necessarily have to be between the schematic parts being displayed diagrammatically.

Golden Gate assembly is a particularly good choice for constructing combinatorial libraries. TeselaGen DESIGN with *j5* can calculate a protocol that will always result in a scarless assembly, even for complex combinatorial libraries. As shown in Fig. 8, every part in each combinatorial bin (the linearized destination vector is

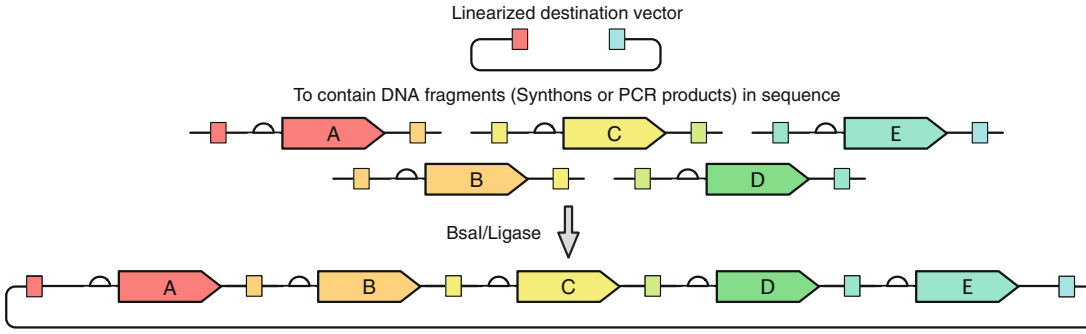

Fig. 7 Golden Gate multipart assembly

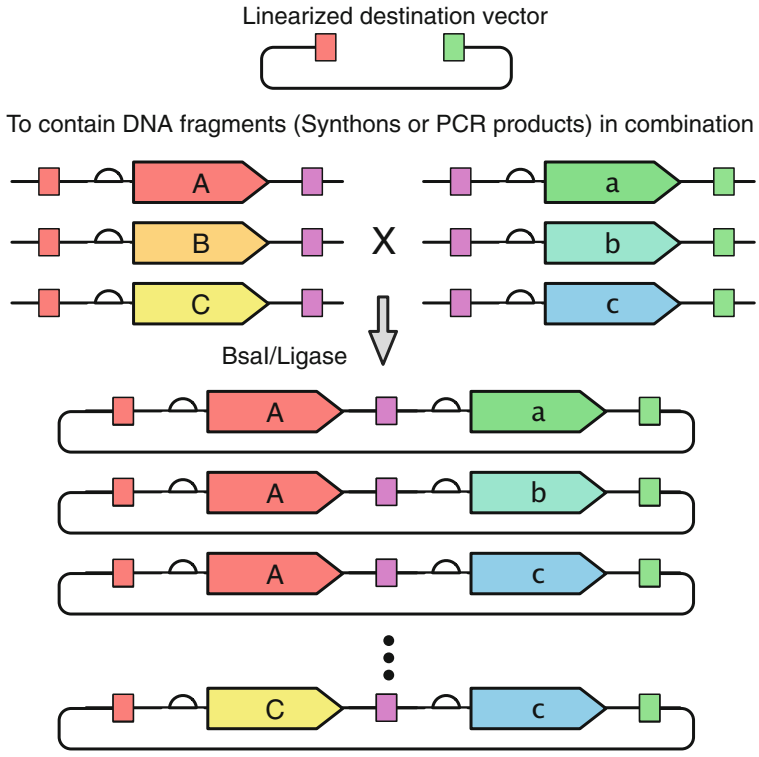

Fig. 8 Combinatorial assembly of a $1 \times 3 \times 3 = 9$ element library

the first bin, the parts labeled A, B, C in the second, and the parts labeled a, b, c in the third) are nominally flanked by the same two 4-bp overhang sequences. Any two parts in a bin are completely interchangeable with respect to Golden Gate assembly, and only a single pair of oligos is required for each part across the entire assembly. If the algorithm cannot find an identical junction across all combinations it relaxes the full reuse constraint (potential for different overhangs for each combination) but never introduces scars (*see* **Note 3**).

2.4 Flanking Homology and Type IIs Endonuclease Method Treatment by TeselaGen DESIGN Module with j5

1. After the user has selected a protocol that maps to the Type IIs assembly method we heuristically determine the most cost-effective strategy to incorporate each part into an assembly fragment prior to executing the full assembly design process. The algorithm calculates the marginal PCR cost and the marginal synthesis cost to make its determination.
   (a) Embed part in primer check. If the part length is less than the minimum allowed part length for PCR, then the part will be embedded in a reverse primer or marked for synthesis as appropriate.
   (b) PCR vs synthesis of a part. Even when synthesis is not chosen as the default strategy, if it is cheaper to synthesize a part, the algorithm will recommend synthesis.
   (c) PCR vs synthesis of next part. If the current part strategy is synthesis, even when synthesis is not chosen as the default strategy for the next part, if it is cheaper to synthesize the next part together with the current part, the algorithm will recommend synthesis.
   (d) PCR vs synthesis of previous part. If the current part strategy is synthesis, even when synthesis is not chosen as the default strategy for the previous part, if it is cheaper to synthesize the previous part together with the current part, the algorithm will recommend synthesis.
2. Progressively relieve violated constraints during primer (or flanking sequence) design. Existing programs such as Primer3 [16] can optimize the design of primers or flanking homology sequences (effectively primers for adjacent assembly pieces during Gibson and CPEC assembly). One drawback is that they provide primer pair designs only if a given set of design criteria is met. This algorithm first attempts to design optimal primers that meet all design constraints; if unable to do so, constraints are progressively relieved until an acceptable primer pair has been found. In addition to the primers (or flanking homology sequences) designed, warning messages are issued if any design constraints were violated/relieved during the design process and/or if any putative template mispriming events with above threshold melting temperatures are identified via BLAST [17].
3. Fragment matching.
   (a) For Flanking Homology: Identify flanking homology assembly piece incompatibilities; if found, design a hierarchical assembly strategy.
      - The algorithm optimizes the flanking homology overlap sequences against typical assembly design parameters.

- The algorithm checks for off target homology annealing sites and designs the assembly process as a 2-level hierarchy if such sites are found. In this way, regions of incompatibly can be buried inside the post first-level contigs input into the second-level assembly process.

(b) For Type IIs: Search for the optimal set of (Golden Gate) assembly piece overhangs.

- For all assembly junctions, the algorithm sets the nominal control position offset to zero. The maximum allowed shift in overhang position is ultimately based on oligo size. As the algorithm attempts for find optimized assembly piece overhangs, shift from neutral will occur, and oligos will be made longer to cover those shifts. As the algorithm proceeds it starts with all neutral offsets, and if any off-target pair of overhangs is incompatible (have too many aligned bps), then shifts the first overhang by +1, and then −1, checking for compatibility across all pairs of overhangs. If not compatible, then that first offset is returned to zero and the next overhang is tested to see if minimalistic offsets can bring the overhangs into a compatible state. Eventually, all combinations of minimal offsets to all overhangs are checked until we find a set that are 100% compatible, or the algorithm determines that no compatible set of overhangs exists given the maximum oligo size.

4. When PCR is called for, closely approximate the optimal distribution of PCR reactions in multiwell plates across thermocycler block annealing temperature zone gradient(s).
5. Checks for DNA fragment buildability and cost are done by submitting DNA fragments to vendors via their APIs. To date, Twist, IDT, and GenScript are developing TeselaGen compatible APIs that provide detailed information to support the design process.

2.5 Other Methods

The list of protocols exploiting modern scarless single-reaction cloning methods is long; however, the number fundamentally different DNA assembly topologies are relatively few. In Table 1 we outline some of the more popular protocols and how they map into a set of topologies that can be used to guide the design of similar protocols using modern DNA assembly software.

Table 1
Examples of industrial cloning methods suitable for designing large-scale DNA libraries

Class	Protocol	Reference	Note
Flanking homology	SLIC	[4]	See text
	Gibson	[5, 11]	See text
	CPEC	[9, 12]	See text
	GeneArt® Seamless	See ThermoFisher website	See text
	In-Fusion®	See Takara website	See text
	Yeast Assembly	[18]	The yeast *Saccharomyces cerevisiae* can take up and assemble at least 38 overlapping single-stranded oligonucleotides and a linear double-stranded vector in one transformation event.
	SLiCE	[10]	See text
	LIC	[19]	Predates SLIC. Inserts are usually PCR amplified and vectors are made linear either by restriction enzyme digestion or by PCR. Technique uses the 3′ → 5′ exo activity of T4 DNA polymerase to create overhangs with complementarity between the vector and insert
Type IIs	Golden Gate	[15]	See text
	MoClo	[20, 21]	See text
	GoldenBraid	[22]	See text
Polymerase/ ligase	DATEL	[23]	DNA assembly method using thermal exonucleases (Taq and Pfu DNA polymerases) and Taq DNA ligase (DATEL)
Blunt end	LCR	[24, 25]	LCR assembly uses single-stranded bridging oligos complementary to the ends of neighboring DNA parts, a thermostable ligase to join DNA backbones, and multiple denaturation–annealing–ligation temperature cycles to assemble complex DNA constructs
Uracil excision	USER	[26, 27]	By varying the design of the PCR primers, the protocol can to perform one or more simultaneous DNA manipulations such as directional cloning, site-specific mutagenesis, sequence insertion or deletion and sequence assembly
Linker mediated	BASIC	[28]	Based on linker-mediated DNA assembly and provides highly accurate DNA assembly
PCR	AFEAP	[29]	The AFEAP method requires two rounds of PCRs followed by ligation of the sticky ends of DNA fragments

3 Example DNA Assemblies Using TeselaGen's DESIGN Module

TeselaGen's DESIGN software provides a unified interface and compute infrastructure for the design of DNA libraries, and the generation of instructions for how to build those DNA libraries (*see* **Note 4**). Those instructions can be generated in a way that can be optimized for both humans and automation. The platform provides a standardized system for tracking the relationships between design elements (parts and annotations), genetic designs (simple, hierarchical, or combinatorial) and DNA assembly protocols. A number of assembly reaction types are supported including Type IIs Endonuclease (Golden Gate, MoClo, etc.), Flanking Homology (Gibson, InFusion, etc.). Among features supporting the design process is the ability to create Design Templates that can be reused across designs. The system optimizes assembly reactions to take advantage of a variety of DNA sourcing options including DNA synthesis vendors such as Twist, IDT, and GenScript.

Important features of the DESIGN module:

- Capture of DNA designs that can be simultaneously combinatorial and hierarchical.
- Scarless design of large-scale DNA libraries.
- Cost optimization including part reuse where warranted.
- Sourcing material from best available options including direct links to DNA vendors.
- Design Templating to aid the design process.
- DNA, Amino Acid, Oligo sequence libraries.
- DNA Combinatorial and Hierarchal Design libraries.
- User lab groups, secure sharing, alerts, and messaging.
- Automated protocol generation for use with automation.

3.1 Design Capture

The fundamental role of the DESIGN module is the accurate capture of the designer's intent. Target genetic designs are constructed 5′ to 3′ (or N-terminus to C-terminus), left to right, by selecting parts from a parts library to columns in a whiteboard style user interface. Alternatives for any given part are listed as cell entries within a column. The user can also specify the naming scheme and the preferred DNA assembly chemical reactions.

3.2 Complex Designs

TeselaGen's DESIGN module has added support for combinatorial DNA libraries as well as hierarchical designs as shown in Fig. 9. The hierarchical capability is particularly useful for the following users:

1. Users who choose to adopt inherently hierarchical assembly methods. They are able to rapidly design complex hierarchical builds using while maximizing part reuse and minimizing cost.

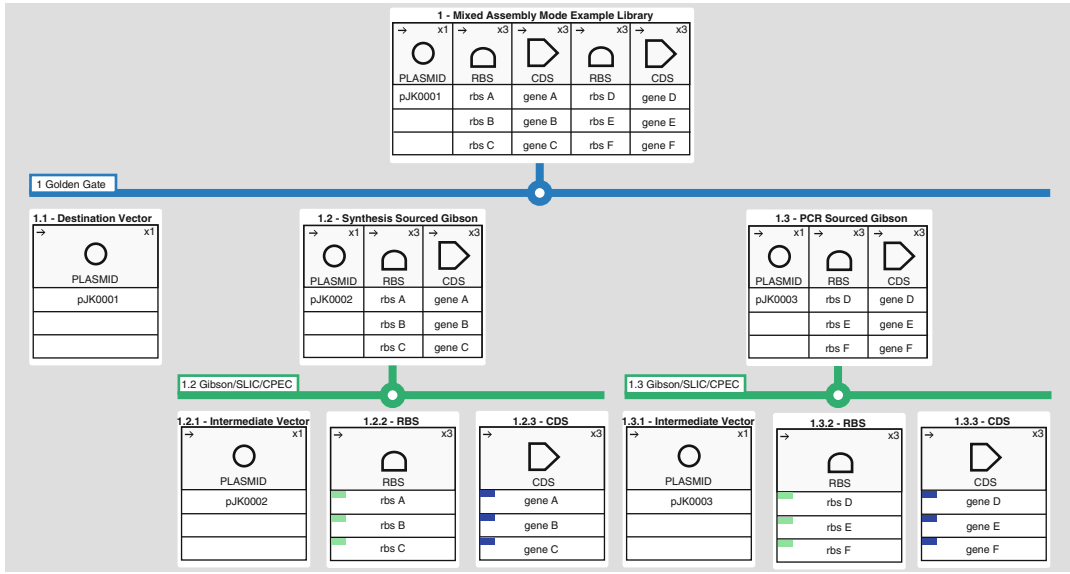

Fig. 9 Hierarchical Design Editor. A two-level combinatorial design specifies four Gibson assemblies that generate intermediates that will be reactants for a final Golden Gate assembly

2. Users who build very long pieces of scarless DNA or construct gene stacks will find the streamlined interface a convenient and reproducible way to break down very long target designs into buildable submodules.

3.3 Part Reuse and Sourcing

Part reuse is important when the size of the built library is large, and cost is a constraint. Without part reuse the cost of a library can grow linearly with the number of parts, while with part reuse it can grow logarithmically. The DESIGN module utilizes the *j5* algorithm when designing a set of combinatorial assembly reactions which automatically maximizes part reuse within a combinatorial design. Hierarchical designs can also be optimized for part reuse across designs with utilities that check if intermediate stretches of DNA already exist in inventory, modifying the build instructions accordingly so that previously built constructs are not assembled again. Users can also use this availability information to automatically break down target constructs into divisions based on available subsections instead of manual divisions. Material availability is can also be extended to query external vendors for what they can provide. This is done through a series of API integrations with vendor utilities that check DNA segments for their manufacturability, cost, and delivery times. The Design module also provides the user with greater control over the sourcing of the DNA parts used in their designs, especially with Type IIs restriction enzyme digest/ligation assemblies. When performing such an assembly, the interface automatically adds validation to the input parts to ensure that they are sourced with the appropriate flanking digest sites.

3.4 Design Rules

The DESIGN module provides support for:

1. Eugene Rules [30], to constrain the complexity of a combinatorial design.
2. Design Rules, with validation logic based on either part tags (e.g., all parts in the first column need to have the "backbone" tag) or a part's base pairs (e.g., all parts in the CDS column need to begin with "ATG").

3.5 Design Templating

The templating system allows users to automate building out portions of designs for complicated hierarchical workflows. Users can capture the common elements of related designs in a template and then apply them across new designs. Any aspect of the design editor can be stored in a template for reuse, including specifications for DNA parts, overhang validation and assembly reactions. This simplifies the design process when creating multiple designs that share characteristics, providing a streamlined interface that minimizes repetitive input from the user.

3.6 Example 1: Combinatorial DNA Library Design

We will use the combinatorial design view of the Design Editor whiteboard to make the design simple and compact. When we are finished with our design it will look something like the design shown in Fig. 10.

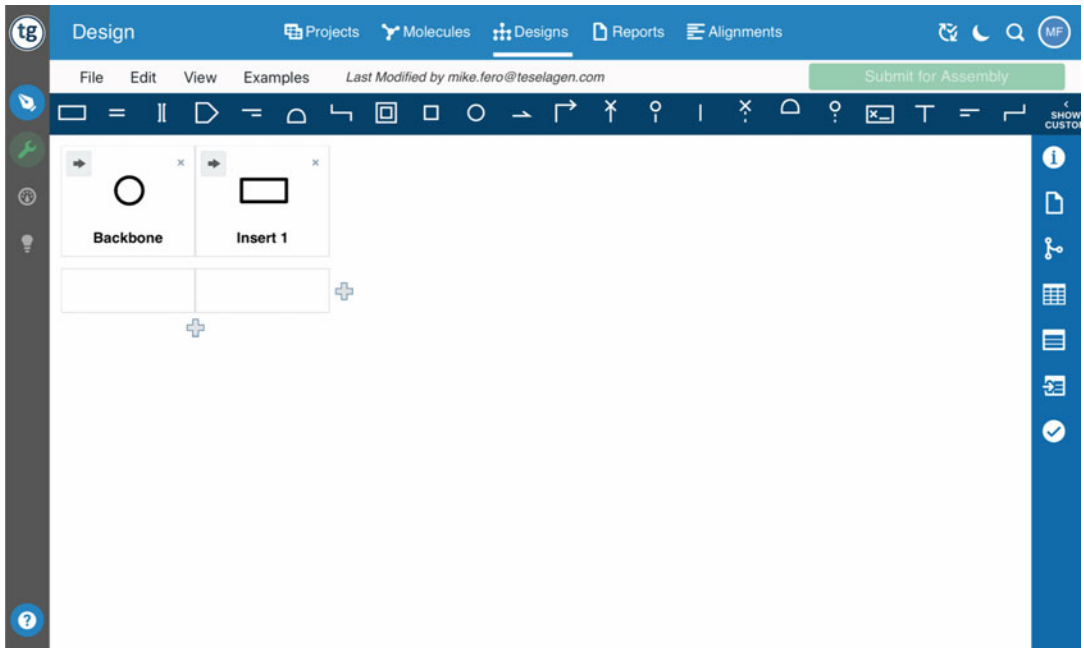

Fig. 10 A combinatorial DNA library design

3.6.1 Sketch Out the DNA Design

We want to divide each construct into the following components:

1. Vector backbone.
2. N-terminal Signal Peptide (2 variants).
3. Gly-Ser Linker (2 variants).
4. GFP (1 variant).
5. ssrA 5-prime degradation tag (2 variants),
6. ssrA 3-prime degradation tag (1 variant).

We will start by creating a new design called Small DNA Library. From the main page, click Designs → Designs → New Design. To add a new component column or variant row, click on the corresponding plus sign. Details of each bin (or component) can be viewed or modified by clicking the bin icon on the inspector panel to the right. Since each construct has six components, let us add four more bins to the current design. In addition, the second, third, and fifth parts each have two variants; therefore, we want to add one more row to the current design.

TeselaGen design editor has a variety of symbols representing different genetic elements that accommodate the needs of visualizing complex DNA constructs. In this design, the second, third, and fourth components are coding sequences; hence, it makes sense to use a "CDS" (coding sequence) icon for these components. For the last two components, we can use a "Protein Stability Element" icon. To change a bin's symbol, click on that bin and choose a symbol from the list of SBOL Visual glyphs [31]. In addition, we can change the name of each component to a more descriptive name by changing its name in the bin's detail tab.

3.6.2 Assign DNA Parts

Now that the design has been sketched, we assign the actual DNA parts. From our DNA Parts Library, we assign the DNA to cells under the proper column headings. Do this by double-clicking a cell you want to assign a DNA part to, the GFP cell for example. A dialog containing the DNA parts library will open up, which allows us to search for "GFP," find the DNA part we are looking for and assign it. For the components with variants (e.g., second, third, and fifth parts), assign the variants to the cells of the same column but on different rows. For example, for the N-terminal signal peptide, the DNA part could be either a BMC_nterm_sig_pep or a ccmN_nterm_sig_pep. Once all the cells have been assigned to actual DNA Parts, the "Submit for Assembly" button turns green and becomes enabled.

3.6.3 Design Rules and Parameters

Here is an important design consideration. The 5-prime and 3-prime degradation tags are rather short. We have chosen to do some PCR to pull some of the desired DNA out of their host vectors, so why not embed these tags into the forward and reverse

primers? We can tell the assembler to do this by assigning a "Forced Assembly Strategy." You can select the forced assembly strategy options for any part by selecting the part of interest, then selecting the part icon on the inspector panel on the right to view the part's details. For our example, let us click select the "ssrA_tag_3prime" cell, then pick "Embed in primer forward" from the "Forced Assembly Strategy" dropdown menu. For the two DNA part options in the ssrA_5primeTag column, pick "Embed in primer reverse." The design editor uses colored markers to indicate different assembly strategies. In our example, the "Embed in primer forward" has the green color bar, while "Embed in primer reverse" has the purple color bar as seen in Fig. 10.

We can add one more feature to this design. It turns out that some of these parts are contiguous in their hosts. The TeselaGen assembler is smart enough not to break everything apart, just to put it all back together again. However, the assembler will do a cost tradeoff between direct DNA synthesis and PCR, based on the cost of DNA synthesis. To tell the assembler not to consider the cost tradeoff and just go with DNA synthesis, you can impose a "Direct Synthesis Firewall" to a column. To do this, select a bin and click on the bin icon on the right panel to view its details. Check the option for "Direct Synthesis Firewall." You should see a red line appear on the right of the selected bin indicating the direct synthesis firewall (DSF) is in effect. In our example we do this for columns 2 and 5. Hence, the red bars in the design shown in Fig. 10.

Finally, you can modify the assembly method by selecting the method from the drop-down menu in the "Assembly Reaction Details." In this design we have chosen Golden Gate as our assembly method.

3.6.4 Submit for Assembly

Navigate to the green "Submit for Assembly" button at the upper right corner of the design editor to run the assembler and build your library.

The DESIGN module does not require you to figure out how to build a library in the traditional artisanal fashion. Once you capture your design in Design Editor, the DNA Assembler takes over and generates all the information you need to build your library. While the DNA Assembler is working a task monitor will appear at the top right of the interface.

3.6.5 Interpret Output

After the DNA Assembler finishes the job, you can view the results by clicking on the Assembly Reports icon on the right panel and navigating to the report of interest. The report starts with information about your design, the assembly method, the time it was run, the export format options, and whether any warning or errors were encountered during the assembly. Following this descriptive metadata, the report is sectioned as follows: prebuilt constructs,

assembled constructs, input sequences, input parts, oligo synthesis, DNA synthesis, PCR reactions, DNA pieces to be assembled, combination of assembled pieces. Let us look at each section in more detail.

1. Prebuilt Constructs: These are constructs that have been built and are available in the library. These are particularly useful for hierarchical designs as they allow us to build complex DNA constructs from simpler constructs that have been built before. In our example, we built our constructs from scratch so there are no prebuilt constructs to display.
2. Assembled Constructs: Recall that we used the combinatorial design editor to create eight constructs. Those are the assembled constructs or output constructs. You can view an assembled construct in the Vector Editor by double-clicking that construct. You can also save the assembled constructs to the DNA sequence library.
3. Input Sequences: As the name suggests, input sequences are the source sequences of the parts in your assembled constructs. They are present in the "DNA Sequences" library.
4. Input Parts: The input parts are the segments of input sequences used in the assembly. They are not necessarily the fragments to be assembled together. This is because the DNA Assembler is smart enough to know not to break up contiguous parts, but to leave them intact to minimize the number of assembly fragments and maximize reuse.
5. Oligo Synthesis: The section tells you what you need to go out and order from your favorite synthesis provider. You have the options to either save it to the oligo library or export as a CSV file. TeselaGen provides direct links to these providers to simplify ordering.
6. DNA Synthesis: This section lists the DNA pieces that need to be directly synthesized with similar properties as the oligos. In our example, there is no direct synthesis.
7. PCR Reactions: This section lists the PCR reactions that need to be done to generate the assembly pieces.
8. DNA Pieces to be Assembled: This section lists the fragments to be put together in the final assembly reactions to give the desired constructs. The parts of each fragment can be viewed from the last section—Combination of Assembly Pieces.
9. Combination of Assembly Pieces: This section lists what goes where to make up the final DNA library. In an automated laboratory setting this list gets translated into a worklist for the robots by the TeselaGen BUILD module, see Fig. 2.

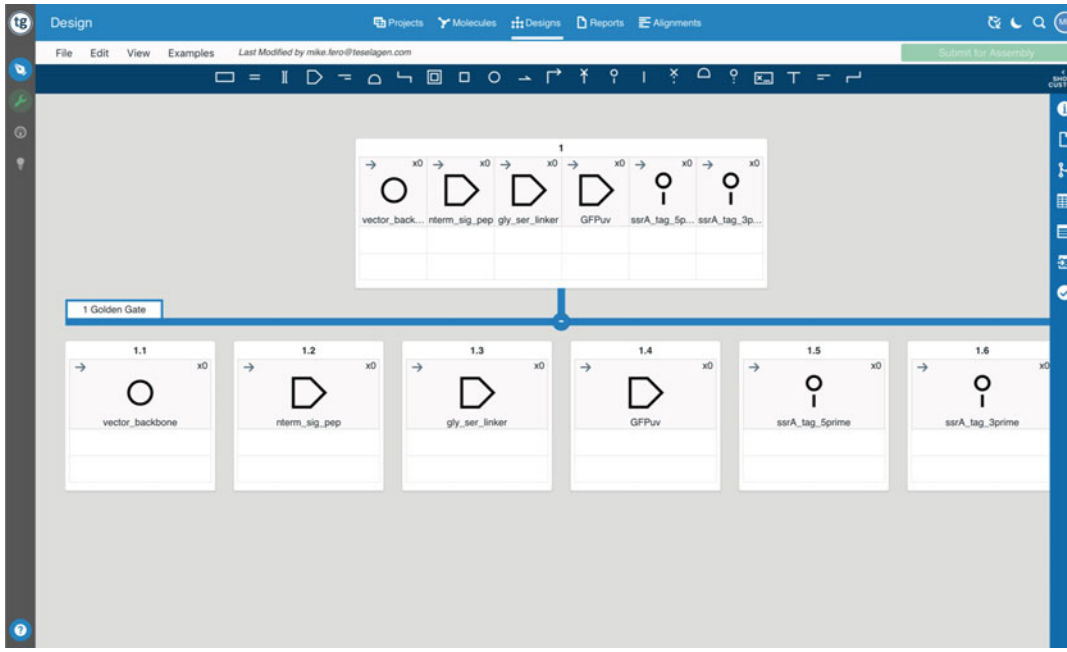

Fig. 11 A hierarchical DNA library design

3.7 Example 2: Hierarchical DNA Library Design

Let us take a look at how to use the TeselaGen DESIGN module to create a DNA library based on a hierarchical approach. In this example we will be use the hierarchical design editor to assemble a library that consists of a backbone, a promoter, 4 ribosome binding sites, and 4 gene using Golden Gate assembly method. Additionally, we want to also specify in the design how the assembly pieces for the final construct was sourced from intermediate constructs via Golden Gate and Gibson assembly. This is a simple design with two levels of hierarchy, but the ideas are extensible to much larger designs with an arbitrary number of levels in the hierarchy. When we have completed the design, it will look something like that shown in Fig. 11.

3.7.1 Sketch Out the DNA Design

We will start by creating a new design called Simple Hierarchical Design. From the main page, click Designs → Designs → New Design. Save the new design as Simple Hierarchical Design. Make sure that you are in an appropriate view mode. From the design editor page, click View → View Mode → Vertical. This view mode separates input and output DNA into their own "cards" with connecting colored lines that symbolize the assembly reaction. The topmost card represents the final DNA assembly product with the assembly reaction beneath it. The tab on the assembly reaction colored lines identifies the reaction. Input cards representing DNA assembly reactants are arrayed below. This process can

continue until DNA fragments that represent the building blocks for the entire hierarchical assembly are defined.

Note that the design paradigm is "top down." Users specify the final assembled construct they would like to build and break that construct into intermediate steps until they reach the lowest level of the hierarchy, the building block parts. Once a design is specified in this way, the software builds from bottom up, generating all of the instructions for the reactions and steps that will result in the desired construct. Note also that there are different ways to approach a hierarchal design. One is a biology centric approach, defining fragments of DNA as "parts" and helping the user think of the design process in terms of functional DNA parts, many of which may already be stored in plasmids and are available to source using simple PCR reactions. Another is a factory approach suitable for service centers or vendors, where the approach might be to just do binary splits of the DNA from the top down until we get to fragment sizes that can be synthesized, then reversing the process through standardized assembly methods. This introductory example favors the biology centric approach in order to stay relevant to most bench scientists.

3.7.2 Layout the Target Construct

When working in the hierarchical view we start with the final goal in mind and then break it down into its constituent parts. For this example, we will want the following bins in our final assembly product card:

1. Backbone (Origin of Replication icon).
2. Promoter (Promoter icon).
3. Ribosome binding site (RBS)(RBS icon).
4. Gene (CDS icon).
5. Ribosome binding site (RBS icon).
6. Gene (CDS icon).
7. Ribosome binding site (RBS icon).
8. Gene (CDS icon).
9. Ribosome binding site (RBS icon).
10. Gene (CDS icon).

Add bins to the topmost card by right-clicking the card and choosing Insert > Insert Bin Right until there are a total of four bins. Next, click on a bin and give it an appropriate icon by clicking the corresponding icon in SBOL glyph ribbon above the design editor canvas. Give the bins a name by clicking on the bin and opening the Bin Details inspector panel from the design editor toolbar along the right side of the screen.

3.7.3 Add First-Level Assembly Reaction

Next, change the assembly reaction to be a Golden Gate reaction instead of the default Mock Assembly (mock assembly does a simple concatenation of sequences in order to make a quick check of the output library). Right click on the reaction tab on the line that connects different elements of the design tree and choose Change Assembly Reaction. From the Assembly Reaction Parameters window, select Golden Gate as the Assembly Method, and Default as the Parameter Preset. Give the reaction a name (e.g., "Golden Gate") and then click the "Next" button. You can also set custom Assembly Report Naming Templates which will affect how the app names various items in the assembly report. Also, if you have a version of this reaction form that you would like to reuse in later designs, you can save the form as an Assembly Reaction Preset. The preset will appear in the reaction preset library and can be used to quickly fill out the reaction form with a single click.

We will be defining the reactants by splitting up the bins of the product card. In our case we want to split the bins in the following way: the promoter, the fourth ribosome binding site (RBS) and the fourth gene are sourced from the vector backbone, each of the other pairs of RBS and gene are sourced together from a different plasmid. To do this splitting, click on the vertical dash lines as shown in Fig. 12 and check the box for "Make all reactants circular (sourced on a backbone)."

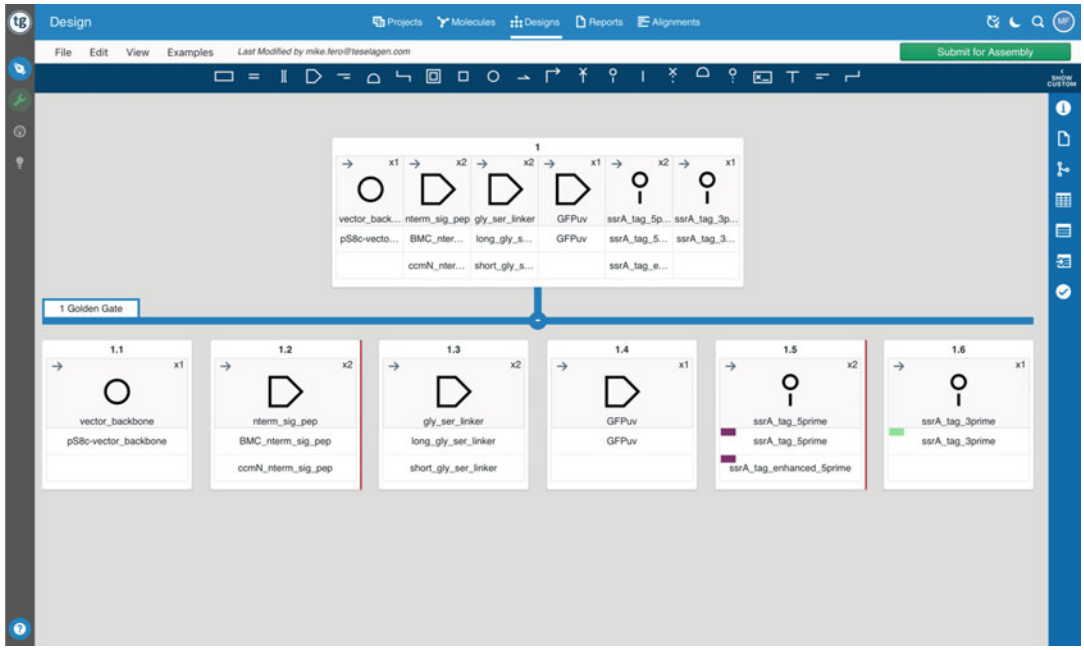

Fig. 12 Assembly reaction definition

3.7.4 Add Second-Level Assembly Reactions

Next let us add our final layer of assembly reactions. This layer describes how each of the intermediate cards (1.1, …, 1.4) are assembled. For example, construct in card 1.1 is assembled via Gibson assembly, while the rests are assembled via Golden Gate. Either right-click the card and choose Add Assembly Reaction or click the [+] button underneath the card. From there, give the reaction a name and assembly method of your choice with the Default Parameter Preset and default Output Naming Templates.

When you are defining the reactant groups, let us choose Split All Bins. Do this for all four cards. You can hide assembly trees by clicking the +/− circle in the middle of a reaction's colored line. This is helpful if you want to focus only on a specific region of a design.

3.7.5 Add Parts to Top Card and Intermediate Cards

Now that the design is specified, we are ready to add DNA parts. In the top Target Construct card, either double click a cell or right-click a cell and choose Insert → Insert Part to add your DNA Parts to the design. Add your own backbone, promoter, genes, and RBS parts in this manner. Once the parts are assigned to the cells in your top card, it is automatically filled in the cells of the cards below. For intermediate cards that contain bins that are not propagated to the top card (e.g., card 1.2, 1.3, and 1.4 have a backbone that is not part of the target card), you need to assign parts for these bins as well. If you had previously imported this example design into your library and are rebuilding it from scratch, then the parts should be in your library. If not, then you may need to import this example design first or use your own data. While a normal Part will usually suit DNA designer's needs, there are several other ways of adding DNA to a design:

- *Part*—An annotation on a source sequence with a start and stop index.
- *Unmapped Part*—A part name that isn't associated with any DNA yet.
- *Base Pairs*—A part unassociated with a sequence consisting of user defined base pairs.
- *Assembly Piece*—A part that already has flanking homology regions on it, when used in a Gibson/SLiC/CPEC reaction our assembly software will design the overlapping ends to conform to the assembly piece part.
- *Sequence*—A convenience method of creating and inserting a part that spans an entire sequence.
- *Part Set*—A group of multiple parts tied together in one UI element, useful to reduce clutter in the design if inserting 100+ parts.

3.7.6 Submit for Assembly

Once all the parts have been inserted, we are ready to Submit for Assembly. The design should look like that shown in Fig. 11. Clicking the green button at top right will send the design to the server to generate the assembly report. Depending on how many combinations are in your design, this process may take a few minutes. If the design is not ready to submit, the green "Submit for Assembly" button will be disabled. You will need at least one part in every bin and every part passing automated validation checks in order to submit to the assembler.

In this design we have five individual assemblies, 4 Golden Gate and 1 Gibson reactions. We will get a separate assembly report for each of these reactions, all linked together in a folder in the Assembly Reports section of the Inspector Panel along the right. Each report can be interpreted individually as explained in Example 1.

3.8 Conclusions

At TeselaGen we are building a four-part AI-guided enterprise platform for bio molecule development that mirrors the design–build–test–evolve ethos of synthetic biology. In this chapter we have described important features of the first of these four modules, the TeselaGen DESIGN module. We have shown that we have been able to direct large scale DNA library construction with the DESIGN module at numerous customer sites, allowing our users to (1) Capture combined and individual combinatorial + hierarchical designs, (2) build large-scale scarless DNA libraries, (3) cost optimize their design/build, (4) source material from best available options including direct links to vendors, (5) allow design templating to aid the "design once, build many" workflows that save time and effort.

Beyond the scope of this chapter, but, nevertheless, interesting to the future of automated and optimized synthetic biology, is the BUILD module that guides communication of protocols to lab workers and automation, the TEST module which gathers high value phenotypic data from analytic instruments, and the EVOLVE module that applies machine learning to optimize workflows, cell line and microbial strain development, and scale-up to commercial production (*see* **Note 5**).

4 Notes

1. Sequence repeats, or highly homologous sequences at the termini of assembly pieces can be problematic for assembly. It may be desirable to include a given part more than once in the same assembly (e.g., a repeated terminator or promoter). However (aside from decreasing the physical stability of the resulting construct (via in vivo recombination processes)), these sequence repeats can be debilitating for the SLIC/Gibson/CPEC/SLiCE assembly methods and should be avoided

where possible. Work-arounds include identify parts with similar function but different DNA sequence (e.g., two sequences encoding the same protein with different codons). If the repeated sequences are not located at the termini of the assembly pieces, they might not significantly affect SLIC or Gibson assembly, but they will be problematic for CPEC or SLiCE assembly. In certain situations, it may be better to perform the assembly with the Golden Gate method, which is not as affected by sequence repeats. *j5* detects highly homologous sequences, automatically alerting the user to these potential problems when designing DNA assemblies.

2. Assembly piece termini with stable secondary structure can be problematic for assembly. If the terminus of an assembly piece has very stable secondary structure (which can be accessed via the DINAMelt Quikfold server, or other related software), as would be anticipated for a terminator, it will not be able to base-pair/anneal with the neighboring assembly piece, and thus inhibit assembly. A work-around is to add sufficient flanking sequence so that the problematic section with secondary structure is no longer at the terminus. In certain situations, it may be better to perform the assembly with the Golden Gate method, which is not as affected by termini with stable secondary structure.
3. For Type IIs Methods (Golden Gate, etc.) there is one situation that will potentially be problematic for assembly: BsaI (or other selected type IIs endonuclease) recognition sites are present within the DNA fragments to assemble (not only at the assembly piece termini). In this case, it is possible to generate (silent) point mutations to disrupt these sites. Even with the undesired BsaI sites present, assembly may still occur (since the digest/ligation is a reversible-process), but the efficiency will be decreased.
4. A variety of detailed protocols compatible with *j5* can be found on the TeselaGen and JBEI websites and Protocols.io.
5. Functional testing is a key bottleneck in high-throughput approaches to screening large libraries for constructs that maximize activity of an enzyme or desired metabolic product. An exhaustive test of a pathway with ten variants across eight parts yields 10^8 possible constructs to be assayed. Obviously, there are two immediate tasks at hand: (1) reduce the complexity of the library and (2) enable building, testing and screening the remaining irreducible set of constructs in the most time and cost-effective way possible. Modern synthetic biology efforts are addressing both these issues, using rules to reduce complexity and automation and intelligent, iterative functional testing to converge using an optimal search routine. It has

proven to be very valuable to integrate design of experiments and statistical learning approaches with construct design/fabrication to minimize the number of constructs screened per round while maximizing information learned.

Acknowledgments

Michael Matena and Adam Thomas for development of TeselaGen's Hierarchical Design Editor; Thomas Rich and Tiffany Dai for development of TeselaGen's Open Vector Editor; Rodrigo Pavez, Tim Thimmaiah, and Nick Elsbree for TeselaGen software platform development to support DNA design; and Katy Basile and Matthew Gibson for intellectual property consultation and support. Portions of this work were funded by NSF 1430986.

References

1. Endy D (2005) Foundations for engineering biology. Nature 438(7067):449–453
2. Arkin A (2008) Setting the standard in synthetic biology. Nat Biotechnol 26(7):771–774
3. Hillson NJ, Rosengarten RD, Keasling JD (2012) j5 DNA assembly design automation software. ACS Synth Biol 1:14–21
4. Li MZ, Elledge SJ (2007) Harnessing homologous recombination in vitro to generate recombinant DNA via SLIC. Nat Methods 4 (3):251–256
5. Gibson DG (2009) Enzymatic assembly of DNA molecules up to several hundred kilobases. Nat Methods 6(5):343–345
6. Chou HH, Hsia AP, Mooney DL, Schnable PS (2004) Picky: oligo microarray design for large genomes. Bioinformatics 20:2893–2902
7. Birla BS, Chou HH (2015) Rational design of high-number dsDNA fragments based on thermodynamics for the construction of full-length genes in a single reaction. PLoS One 10(12): e0145682
8. Li L, Jiang W, Lu Y (2018) A modified Gibson assembly method for cloning large DNA fragments with high GC content. Methods Mol Biol 1671:203–209
9. Quan J, Tian J (2009) Circular polymerase extension cloning of complex gene libraries and pathways. PloS one 4(7):6441
10. Zhang Y, Werling U, Edelmann W (2012) SLiCE: a novel bacterial cell extract-based DNA cloning method. Nucleic Acids Res 40 (8):e55
11. Ramon A, Smith HO (2011) Single-step linker-based combinatorial assembly of promoter and gene cassettes for pathway engineering. Biotechnol Lett 33(3):549–555
12. Quan J, Tian J (2011) Circular polymerase extension cloning for high-throughput cloning of complex and combinatorial DNA libraries. Nat Protoc 6(2):242–251
13. Engler C, Gruetzner R, Kandzia R, Marillonnet S (2009) Golden Gate shuffling: a one-pot DNA shuffling method based on type IIs restriction enzymes. PLoS One 4(5):e5553. https://doi.org/10.1371/journal.pone.0005553
14. Engler C, Marillonnet S (2011) Generation of families of construct variants using golden gate shuffling. Methods Mol Biol 729:167–181
15. Engler C, Kandzia R, Marillonnet S (2008) A one pot, one step, precision cloning method with high throughput capability. PLoS One 3 (11):3647
16. Rozen S, Skaletsky H (2000) Primer3 on the WWW for general users and for biologist programmers. Methods Mol Biol 132:365–386
17. Zhang Z, Schwartz S, Wagner L, Miller W (2000) A greedy algorithm for aligning DNA sequences. J Comput Biol 1(1-2):203–214
18. Gibson DG (2009) Synthesis of DNA fragments in yeast by one-step assembly of overlapping oligonucleotides. Nucleic Acids Res 37 (20):6984–6990
19. Aslanidis C, de Jong PJ (1990) Ligation-independent cloning of PCR products (LIC-PCR). Nucleic Acids Res 18 (20):6069–6074
20. Weber E, Engler C, Gruetzer R, Werner S, Marillonnet S (2011) A modular cloning

system for standardized assembly of multigene constructs. PLoS One 10(1371):0016765

21. Werner S, Engler C, Gruetzner R, Marillonnet S (2012) Fast track assembly of multigene constructs using Golden Gate cloning and the MoClo system. Bioeng Bugs 3(1):38–43
22. Sarrion-Perdigones A, Falconi E, Zandalinas S, Juárez P, Fernández-del-Carment A, Granell A, Orzaez D (2011) GoldenBraid: an iterative cloning system for standardized assembly of reusable genetic modules. PLoS One 6(7): e21622
23. Jin P, Ding W, Du G, Chen J, Kang Z (2016) DATEL: a scarless and sequence-independent DNA assembly method using thermostable exonucleases and ligase. ACS Synth Biol 5 (9):1028–1032
24. de Kok S, Stanton LH, Slaby T, Durot M, Holmes VF, Patel KG, Platt D, Shapland EB, Serber Z, Dean J, Newman J, Chandran SS (2014) Rapid and reliable DNA assembly via ligase cycling reaction. ACS Synth Biol 3 (2):97–106
25. Chandran S (2017) Rapid assembly of DNA via ligase cycling reaction (LCR). Methods Mol Biol 1472:105–110
26. Bitinaite J, Rubino M, Varma KH, Schildkraut I, Vaisvila R (2007) USER™ friendly DNA engineering and cloning method by uracil excision. Nucleic Acids Res 35 (6):1992–2002
27. Bitinaite J, Nichols NM (2009) DNA cloning and engineering by uracil excision. Curr Protoc Mol Biol Chapter 3:Unit 3.21
28. Storch M, Casini A, Mackrow B, Ellis T, Baldwin GS (2017) BASIC: a simple and accurate modular DNA assembly method. Methods Mol Biol 1472:79–91
29. Zeng F, Zang J, Zhang S, Hao Z, Dong J, Lin Y (2017) AFEAP cloning: a precise and efficient method for large DNA sequence assembly. BMC Biotechnol 17(1):81
30. Bilitchenko L, Liu A, Cheung S, Weeding E, Xia B, Leguia M, Anderson JC, Densmore D (2011) Eugene – a domain specific language for specifying and constraining synthetic biological parts, devices, and systems. PLoS One 6(4):e18882
31. Quinn J, Cox RS, Adler A, Beal J, Bhatia S, Cai Y, Chen J, Galdzicki M, Clancy K, Hillson N, Le Novère N, Maheshwari A, McLaughlin JA, Myers C, Umesh P, Pocock M, Rodriguez C, Soldatova L, Stan G-BV, Swainston N, Wipat A, Sauro HM (2015) SBOL visual: a graphical language for genetic designs. PLoS Biol 13(12):e1002310