


# AssemblyTron: flexible automation of DNA assembly with Opentrons OT-2 lab robots

John A. Bryant Jr., Mason Kellinger, Cameron Longmire, Ryan Miller, and R. Clay Wright 

Department of Biological Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

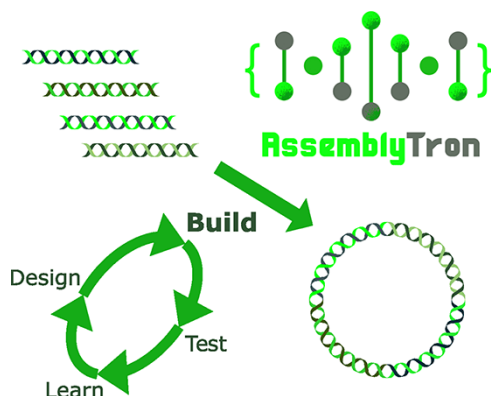
\*Corresponding author: E-mail: [wrightrc@vt.edu](mailto:wrightrc@vt.edu)

## Abstract

As one of the newest fields of engineering, synthetic biology relies upon a trial-and-error Design–Build–Test–Learn (DBTL) approach to simultaneously learn how a function is encoded in biology and attempt to engineer it. Many software and hardware platforms have been developed to automate, optimize and algorithmically perform each step of the DBTL cycle. However, there are many fewer options for automating the build step. Build typically involves deoxyribonucleic acid (DNA) assembly, which remains manual, low throughput and unreliable in most cases and limits our ability to advance the science and engineering of biology. Here, we present AssemblyTron, an open-source Python package to integrate j5 DNA assembly design software outputs with build implementation in Opentrons liquid handling robotics with minimal human intervention. We demonstrate the versatility of AssemblyTron through several scarless, multipart DNA assemblies, beginning from fragment amplification. We show that AssemblyTron can perform polymerase chain reactions across a range of fragment lengths and annealing temperatures by using an optimal annealing temperature gradient calculation algorithm. We then demonstrate that AssemblyTron can perform Golden Gate and homology-dependent *in vivo* assemblies (IVAs) with comparable fidelity to manual assemblies by simultaneously building four four-fragment assemblies of chromoprotein reporter expression plasmids. Finally, we used AssemblyTron to perform site-directed mutagenesis reactions via homology-dependent IVA also achieving comparable fidelity to manual assemblies as assessed by sequencing. AssemblyTron can reduce the time, training, costs and wastes associated with synthetic biology, which, along with open-source and affordable automation, will further foster the accessibility of synthetic biology and accelerate biological research and engineering.

**Key words:** laboratory automation; DNA assembly; molecular cloning; design cycle; robotic liquid handling

## Graphical Abstract



## 1. Introduction

As one of the newest fields of engineering, biological engineering or synthetic biology is still largely performed by iterative, trial-and-error Design–Build–Test–Learn (DBTL) cycles, because both our knowledge of biology is incomplete and our ability to build biological devices is error-prone. Molecular biology developments

in the past 40 years have improved our ability to treat genes, promoters and other deoxyribonucleic acid (DNA) elements as modular parts to engineer increasingly complex devices with biology. However, assembly of the DNA elements encoding these devices largely remains error-prone and requires significant tacit knowledge and iterative trials to master (1). The speed at which we can

Submitted: 15 September 2022; Received (in revised form): 25 October 2022; Accepted: 21 December 2022

© The Author(s) 2023. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

engineer biology and learn its design principles relies on our ability to perform DBTL cycles rapidly and error-free. Automation of the build step can accelerate each DBTL cycle, eliminate errors and allow scientists and engineers to focus their energy on the creative design and learn steps addressing the biological questions they set out to answer instead of optimizing the physical assembly process of the build step. Notably, a 2017 analysis of biological design automation tools found that there were fewer automation tools available for the Build step than Design, Test or Learn (2).

To build biological devices, researchers assemble multigene DNA constructs using an ever-growing variety of potential DNA assembly methods, including short homology recombination methods such as SLIC (3), Gibson (4), CPEC (5), AQUA (6) and *in vivo* assembly (IVA) (7), restriction–ligation methods such as Golden Gate (8) and *de novo* synthesis (9, 10). Each of these methods has advantages and disadvantages, and a combination of short homology recombination, restriction–ligation and *de novo* synthesis methods is routinely used in most synthetic biology laboratories. Importantly, flexible combinatorial fragment assembly methods allow the reassembly of different parts (e.g. various promoters and terminators with a coding sequence) for creating combinatorial construct libraries with high time and cost efficiency increasing DBTL cycle throughput. Thus, even as the cost of *de novo* DNA fragment synthesis continues to decrease, the need will remain for robust, automated assembly capabilities in order to accommodate higher throughput experimentation.

Build automation is a key to improving the speed and accuracy of DNA assembly and accelerating the general DBTL cycle, as human errors are frequent in manual assembly processes. The Edinburgh Genome Foundry has automation workflows that increase throughput 20-fold, while iBioFab at the University of Illinois at Urbana-Champaign has reduced the price of construct assembly by 97.7% (11, 12). Although these impressive figures underpin the value of biofoundries, high experimental costs still prevent most of the academic community from taking advantage of these facilities or incorporating automation into their laboratories. Additionally, standard procedures in established biofoundries including software, hardware and methods are not interoperable, so their services are often useful to only a small pool of researchers (13). Although a flexible, open-source build automation platform would be a good alternative, few exist (14, 15), particularly those that make use of liquid handling robotics. Many of the build platforms that do exist are fairly inaccessible, difficult to find and not interoperable (2, 16). This hampers the efficiency of DBTL cycles by preventing remote design implementation and automatic data feedback for optimized *in silico* redesign (17).

Recently, low-cost, open-source liquid handling robotics systems have created an opportunity to increase the throughput and decrease the human error associated with the build step of the DBTL cycle (18). The Opentrons OT-2 is an open-source liquid handling robot with thermocycler capabilities. Its advanced Application Programming Interface library for pipette manipulation and protocol development is highly flexible for a wide array of molecular biology applications. However, there is not yet an open-source Python software package or protocols to enable the OT-2 to execute the popular short homology and Golden Gate DNA assembly protocols, such as those generated by automated DNA assembly design algorithms such as j5 and Cello (19, 20). To our knowledge, the only build automation software for the OT-2 is DNA-BOT (21), which executes the Biopart Assembly Standard for Idempotent Cloning method (22).

Numerous options exist for researchers to automate the generation of DNA device designs and optimize assembly protocols, which implement commonly used, scarless DNA assembly methods, but these often do not integrate with liquid handling robotics. To minimize researcher-to-researcher variation in primer and assembly design and maximize the likelihood of assembly success, several software platforms have been developed to automate the design and optimization of DNA assemblies using vetted algorithms (19, 20). The j5 construct design algorithm is a valuable tool for standardizing the design and assembly workflow while minimizing the need for DNA synthesis (19). The interpretation and implementation of the DNA assembly as specified in j5 output remain a bottleneck in the DBTL cycle. There is a significant amount of training required for new researchers to master DNA assembly and a high potential for errors during training. This training barrier often prevents undergraduates from being successful in their first synthetic biology experiments. Software for processing output files from DNA assembly design software, such as j5, and generating protocols for this new generation of relatively affordable laboratory robotics, such as the OT-2, would expedite automation efforts in public laboratories, avert errors on the benchtop and allow researchers to focus on the more critical tests and learn steps.

We seek to minimize the human error rate in the build step of the DBTL cycle by automating the physical assembly workflow with open-source software that implements j5 assembly designs in the OT-2 liquid handling robotics system with minimal human intervention. We aimed to create a build automation platform that supports existing DNA assembly protocols and conventions. We also focused on leveraging existing DNA design software and liquid handling robotics hardware platforms to expedite synthetic biology across academic research laboratories in an economically accessible way. Here, we present AssemblyTron, an open-source Python package to implement DNA assembly design software output (currently j5 specifically) using the OT-2 liquid handling robot (23). This package is designed to automate the DNA assembly process between the design stage and transformation in *Escherichia coli* with minimal human intervention. AssemblyTron automates enzyme-free homology-dependent assembly methods such as AQUA (6) and IVA (7), which serve as fast and cheap alternatives to Gibson since they depend on native *E. coli* machinery to ligate fragments. It also automates Golden Gate (8) assembly to offer a highly accurate and flexible assembly strategy option for complex designs or libraries. AssemblyTron is an open-source build automation framework that can reduce the time, training, costs and excesses associated with molecular biology workflows. This allows researchers to spend more time asking questions and building new synthetic biology tools instead of troubleshooting their assembly workflow.

## 2. Materials and methods

### 2.1 *Escherichia coli* strains and reagents

Media and other reagents were prepared according to Green et al. (2012), unless otherwise specified (24). *Escherichia coli* TOP10 chemically competent cells were prepared by the Hanahan method (25) (efficiency of  $8.47 \times 10^8$  colony-forming units per  $\mu\text{g}$  (CFU/ $\mu\text{g}$ ) supercoiled pUC19). Lysogeny broth (LB), Miller (Fisher BioReagents) medium with kanamycin or ampicillin at 50 or 100  $\mu\text{g}/\text{mL}$ , respectively, was used for growing *E. coli* with the addition of 0.7% (w/v) bactoagar for plating. The New England Biolabs (NEB) Monarch Plasmid MiniPrep kits were used for isolating plasmid

DNA from *E. coli*, and the Zymogen DNA Clean and Concentrate kits were used to purify polymerase chain reaction (PCR) products prior to assembly or transformation. Agarose gels (1% (w/v) in 1× Tris-acetate-EDTA) were stained with Biotium GelRed® Nucleic Acid Gel Stain, and bands were imaged using an iBright imaging system (Thermo Fisher). The 1 kb plus DNA ladder (NEB) was used for fragment length comparison via agarose gel electrophoresis.

## 2.2 Primers and plasmids

Primers were designed using j5 (19) and were purchased from Integrated DNA Technologies (Table S1 and File S7). j5 is free for use in academic laboratories (j5.jbei.org). Plasmids used included pGP8A-ARF19 (gifted from the Nemhauser Lab), pGP8A-ARF5-pdar, pGP8A-ARF7-pdar, pIDMv5K-J23100-tsPurple-B1006, pIDMv5K-J23100-YukonOPF-B1006, pIDMv5K-J23100-aeBlue-B1006 and pIDMv5K-J23100-fuGFP-B1006 (gifted from Sebastian Cocioba) (Table S2 and File S8).

## 2.3 DNA fragment construction and plasmid assembly

PCRs were performed in 25 µL volumes using Phusion® High-Fidelity DNA Polymerase (NEB) or Q5® High-Fidelity DNA Polymerase (NEB) with 0.1 µM primers and 0.5 ng linearized plasmid template DNA. Template plasmids were linearized with restriction enzymes cutting outside of the desired amplicons to improve amplification. Amplification was performed according to the following protocol: 30s at 98°C, 34 or 36 cycles of 10s at 98°C, 30s at annealing temperatures specified by AssemblyTron/j5, extension times as specified by AssemblyTron at 72°C and a final 5 min extension at 72°C. For plasmid template digestion, 19 µL water, 5 µL rCutSmart Buffer (NEB) and 1 µL DpnI (NEB) were added and incubated for 30 min at 37°C prior to deactivation at 65°C for 20 min.

Final assemblies for Golden Gate (Figure 3) and fragment mixes for homology-dependent assembly (Figure 5) were cleaned and concentrated with DNA Clean and Concentrator-5 columns (Zymo), and DNA was eluted with 10 µL molecular grade water. This elution was transferred into 50 µL of TOP10-competent cells. The other 5 µL was used to measure DNA concentration for transformation efficiency calculations with a NanoDrop-2000c Microvolume Spectrophotometer (Thermo Fisher). Transformation mixes were incubated for 30 min on ice. Mixes were then heat shocked at 42°C for 60s and recovered at 37°C for 60 min with 250 µL LB with catabolite repression (LB + 0.2% (w/v) dextrose) medium added. Depending on the predicted assembly efficiency, 50–200 µL of the base mix or a 10× dilution was plated onto LB agar plates with the appropriate antibiotics for incubation at 37°C overnight. Colonies were manually counted and reported as CFU/µg of DNA plated. Plasmid assembly was assessed by chromoprotein expression, Sanger sequencing aligned with A Plasmid Editor (26) and whole plasmid sequencing (Plasmidsaurus).

Fragment mixes for Figure 4 were transformed into a separate batch of TOP10-competent cells and were recovered with Super Optimal Broth with catabolite repression. The efficiency of these transformations was comparable to those above; however, as these were performed with a separate batch for competent cells transformation efficiencies in Figure 4 should not be directly compared with those in Figures 3 and 5.

## 2.4 Liquid handling protocol

DNA constructs were designed using j5 (Files S1–S4) (Figure 1A). Following design, a Python setup script was run on the Anaconda

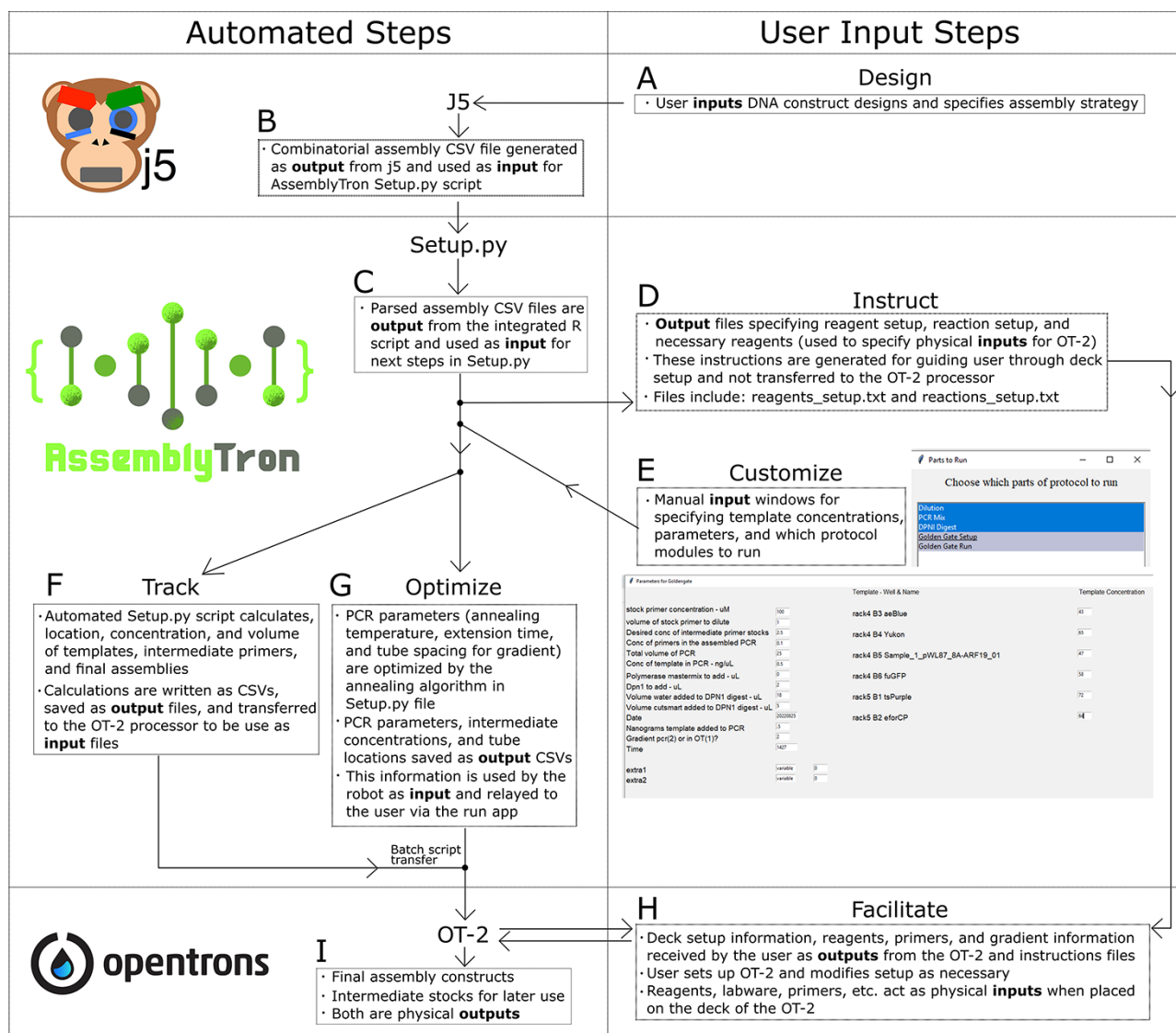
command prompt to launch AssemblyTron. First, the setup script used a subprocess to call an R script included in AssemblyTron to parse the combinatorial j5 designs (Figure 1B) into four CSV files: oligo.csv, assembly.csv, combinations.csv and pcr.csv (Figure 1C). Parsed CSV files were then automatically transferred from our shared Google drive to the AssemblyTron working directory. The setup script then generated a custom text file with instructions on how to set up the deck for the current protocol (Table S3) (Figure 1D). Next, it prompted a pop-up window for specifying template concentrations, modifying PCR volumes, etc., and saved these parameters as another CSV in the working directory (Figure 1E). The setup script then ran the optimal annealing algorithm, saved the gradient as a CSV file and attached tube positions to the pcr.csv file (Figure 1G). Finally, AssemblyTron calculated the location, concentration and volume of reagents and DNA and saved the information as CSV input files (Figure 1F). These files are archived in the working directory following the run. It then made copies of the automated dilution and protocol scripts and produced another instructions file for positioning PCR tubes in the thermocycler (Table S4). A batch script was then called via a subprocess from the setup script to copy all CSV input files to the robot working directory. This batch script is included in the package download. The resulting protocol scripts in the working directory were then uploaded to the OT-2 run app where they were thoroughly screened and debugged. The AssemblyTron Golden Gate script is based on Engler et al. (8), and the homology-dependent assembly script is inspired by García-Nafria et al. (7) and Beyer et al. (6, 24). PCRs were manually transferred to a Bio-Rad C100 gradient thermocycler. Cycles were performed as stated previously, and the reactions were then returned to the OT-2 for assembly (Figure 1H). After assembly, final constructs and intermediate stocks were collected from the OT-2 deck (Figure 1I). A vignette intended to guide a novice user through the installation and use of AssemblyTron is included in File S9.

## 3. Results

To demonstrate the utility and performance of the AssemblyTron package, we have used it to build two sets of combinatorial plasmid assemblies: one consisting of two yeast expression vectors containing plant transcription factors assembled from two fragments and another consisting of four chromoprotein *E. coli* expression vectors assembled from four fragments each. The chromoprotein assemblies enable high-throughput quantification of the efficiency and accuracy of assembly.

To use AssemblyTron, the user must first begin with a combinatorial DNA construct design and select an appropriate assembly strategy (Figure 1A). This information is then used as input for the j5 algorithm (19), which generates combinatorial assembly files (Figure 1B). Combinatorial assembly files specify the primer sequences, PCR parameters and fragment assembly strategy. Several combinatorial assembly files for different sets of constructs from different users can be combined with j5 tools to batch many assemblies across a laboratory or biofoundry user base. AssemblyTron in its current iteration is limited to assemblies with a maximum of 96 total primers and templates.

AssemblyTron then converts the combinatorial design files into Opentrons protocols and instruction documents for the operator. A single combinatorial assembly file from j5 is used as the input to the AssemblyTron package. The user begins by initiating the Setup.py script in AssemblyTron, which facilitates all file processing. The combinatorial assembly file is first parsed by an AssemblyTron R script called by Setup.py into four individual CSV files:



**Figure 1.** AssemblyTron workflow: (A) Design: user must devise a DNA construct design and choose the appropriate assembly strategy (Golden Gate, Gibson, etc.). (B) After making the design in j5, the user receives a combinatorial design file as an output, which is used as an input for AssemblyTron. (C) The AssemblyTron Setup.py script runs an integrated R script to parse the j5 combinatorial design file and divides it into separate CSV files for different stages of assembly. This R script runs automatically in the Setup.py script. (D) Instruct: after initiating the Setup.py script and specifying the location of the parsed design files, the user receives a reagent\_setup file as an output (Table S3). This file relays the deck setup to the user and specifies the arrangement of primers, templates, etc., on the OT-2 deck, which we term physical inputs. (E) Customize: the user is manually prompted to input template concentration, which parts of the protocol to run, etc. (F) Track: Setup.py calculates and tracks the location, concentration and volume of all primers, templates and final assemblies. This information is provided as CSV output files, which are provided for user reference and input for the OT-2. (G) Optimize: AssemblyTron optimizes the annealing temperature and extension time of PCRs with its optimal annealing algorithm. Instructions for arranging 100  $\mu$ L PCR tubes in the OT-2 thermocycler are provided in reactions\_setup.txt. (H) Facilitate: the user stages the OT-2 deck and adjusts it as necessary according to instructions text files and prompts from the run app. (I) The user is left with final assembly constructs as well as intermediate primer and template stocks, which we term physical outputs.

oligo.csv, pcr.csv, assembly.csv and combinations.csv (Figure 1C). The script then moves the new CSV files to the AssemblyTron working directory and processes them to generate an instructions file (Figure 1D), which instructs the user which reagents to retrieve and how to set up the OT-2 deck for the run. Next, Setup.py prompts the user to input the concentrations of template stocks for dilutions (Figure 1E). Other parameters, such as PCR volumes and primer concentrations, can be modified but are prefilled for convenience. Additionally, there are two slots in the graphical user interface for extra parameters in the case that a user modifies the source code.

Following parameter confirmation, Setup.py calculates concentrations for working stocks of templates, primers, intermediates and final products for the run. This means that AssemblyTron tracks and records every item through each step of the reaction and saves the information as CSV files (Figure 1F), which could in the future interface with laboratory inventory management systems. Setup.py then transfers protocol scripts to the OT-2 via a batch script (Figure 1F and G). Setup.py also defines an optimal thermocycler gradient and block position (annealing temperature) for each PCR and an optimal extension time for the set of reactions (Figure 1G). Since this optimization algorithm

relies on the implementation of a gradient annealing step, the current OT-2 thermocycler module is insufficient for our workflow. The user must manually transfer PCR tubes to a separate thermocycler and manually enter the PCR parameters; however, PCR tube spacing and gradient parameters are calculated and given by AssemblyTron in the reaction instructions text file. Alternatively, during the assembly design steps in j5, the user may specify stringent primer parameters (i.e. Primer Max T<sub>m</sub> Diff, Primer Max T<sub>m</sub> and Primer Min T<sub>m</sub>), although this method is less robust to difficult sequences. PCR products may either be purified or directly returned to the OT-2, and the assembly protocol is performed. These manual steps, which are necessary for facilitating automation with the OT-2, are depicted in Figure 1H. At the conclusion of the OT-2 run, the user is left with assembled constructs to be transformed, remaining intermediate dilutions for later use (Figure 1I) and CSV files from the Track step (Figure 1F), in which each step of the protocol and location of reagents is tracked.

### 3.1 PCR

First, we demonstrate the software's ability to automatically prepare PCRs directly from input files (Files S1–S4). AssemblyTron is currently capable of performing PCRs with up to 96 combined primers and templates in standard microcentrifuge tubes, which are diluted into a 96-well plate. Our script first calculates an optimal annealing temperature gradient that accommodates the annealing temperature of each fragment for assembly <math><0.4^{\circ}\text{C}</math>. j5 also includes a 'delta' parameter, which specifies the tolerable variance from primer annealing temperatures for each PCR. If the delta parameter is <math><0.4^{\circ}\text{C}</math>, the script will adjust the gradient to where these more rigid annealing temperatures are accommodated. The annealing temperature gradient accommodates each PCR in a single run. A reaction tube number is assigned to each temperature step in the gradient in order to specify tube placement (Table S4). These PCR tube positions are relayed to the user via a pop-up text file. The script uses the gradient to choose the tube positions, and once reactions are mixed, they are ready for gradient amplification. The OT-2 thermocycler module does not currently have gradient capability, so it is necessary to move mixed reactions into a separate gradient thermocycler to run the reaction. However, all calculations, optimization and setup for the run are determined by our software package. This reduces the cognitive burden of molecular cloning, reduces the rate of failure and eliminates the need for any manual protocol optimization. Finally, our package prompts the Opentrons user interface to

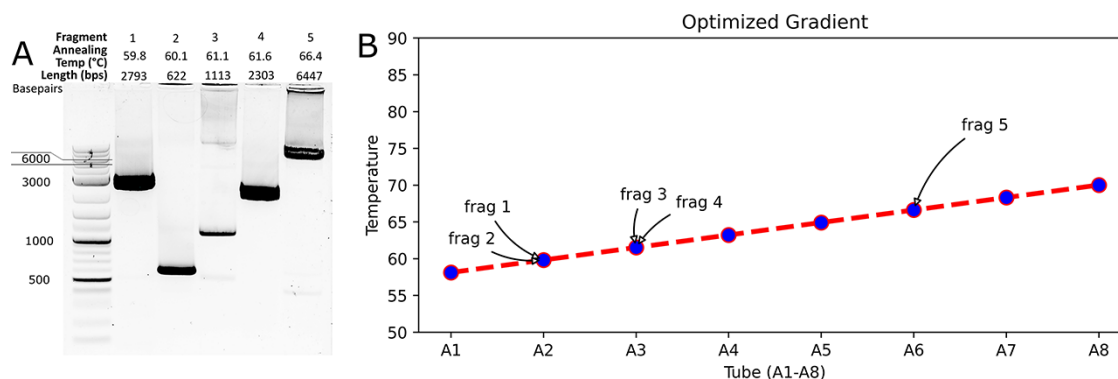
pause and instructs the user on how to set up the gradient PCR run. The main drawback of PCR is the trial-and-error optimization process, and our package harnesses machine precision to automate and resolve PCR parameter calculation using j5 design output files.

To demonstrate our PCR script and gradient algorithm, we amplified five fragments with variable sizes and calculated the means and differences annealing temperature (Figure 2A). Our setup.py script contains the gradient algorithm, working directory generation code and parameter input window for staging the protocol. Input files are found in Files S1–S4.

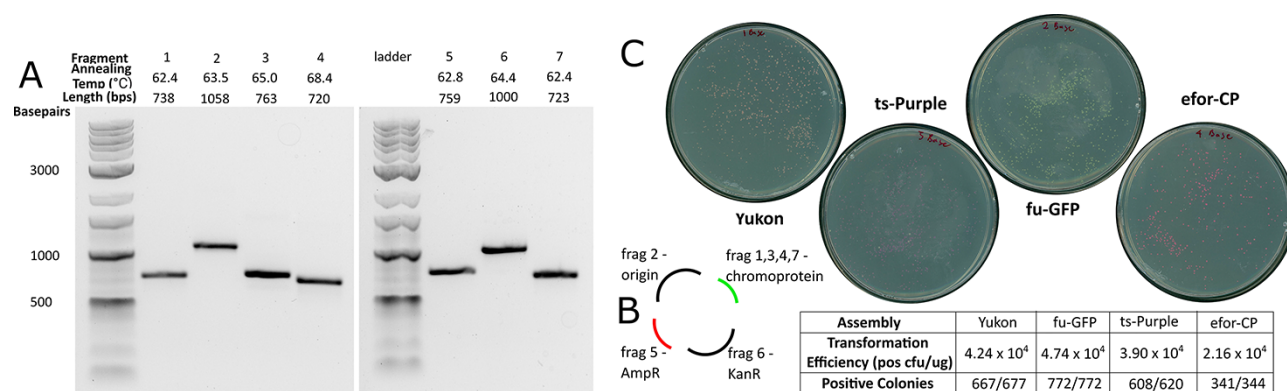
Gel electrophoresis of the final PCR products indicated that we successfully amplified all fragments and obtained the expected sizes from a single gradient thermocycler run (Figure 2A). It should be noted that Fragment 5 had a significantly higher annealing temperature (>4°C) than the other four fragments. To demonstrate how our algorithm accommodates this difference, Figure 2B provides a visual plot of the optimal linear temperature gradient calculated by our algorithm. The algorithm determined that Fragment 5 should be in a tube located in Row A at Column 6, which has a higher temperature gradient than the other fragments. This information was then relayed to the robot, which assembled the reaction mix in the correct position. The reactions are transferred to the thermocycler, and the gradient cycle was set as outlined in the instructions document (Table S4). All reactions were performed successfully, as shown in Figure 2A.

### 3.2 Golden Gate assembly

AssemblyTron also includes a Golden Gate assembly feature, which allows the user to go from primers and templates specified in a j5 Golden Gate assembly file to fully assembled, transformation-ready plasmids in a single day. This feature begins by performing a gradient-optimized PCR run for each fragment in separate tubes. Primers and templates specified from the j5 design files are used for the PCR run (Figure 3A). Next, a Dpn1 digestion is performed to remove the residual template. Our package then prompts the robot to automatically combine amplified fragments in equimolar ratios for each assembly reaction, assuming equivalent yields from each PCR. If any reactions of shorter fragments are likely too concentrated, a dilution will be performed to accommodate accurately pipettable equimolar additions of each fragment. Next, reagents are added prior to automatic initiation of the assembly in the Opentrons thermocycler module. The Opentrons thermocycler module is practical for Golden Gate assemblies since there is no parameter gradient required across



**Figure 2.** PCR with AssemblyTron. (A) AssemblyTron PCR protocol was verified by successful amplification of DNA fragments with variable annealing temperatures and lengths. (B) A graphical depiction of the optimal annealing temperature gradient calculated for the PCR. Each tube is assigned a temperature in the linear gradient, which corresponds to an annealing temperature of one or more fragments.



**Figure 3.** Golden Gate assembly with AssemblyTron. (A) AssemblyTron protocol was verified by successful amplification of each chromoprotein assembly fragment. (B) A schematic diagram to specify how fragments are assembled to yield final constructs. (C) High transformation efficiency, high accuracy and correct chromoprotein expression validate the robustness of the Golden Gate protocol.

reactions, in contrast to PCR. Although there is no need to relocate the reaction to a separate thermocycler, the protocol will pause so that reactions can be transferred in case the laboratory does not have access to the Opentrons thermocycler module. Once the Golden Gate thermocycler protocol is completed, the reactions can be removed, optionally cleaned and concentrated and transformed into *E. coli*. Golden Gate assembly reactions performed using AssemblyTron are accurate and efficient (Figure 3D).

To test the AssemblyTron combinatorial Golden Gate assembly feature, we also assembled chromoprotein *E. coli* expression vectors using this method. The assembly consists of four final plasmids, which were each assembled from four fragments. The fragments include two separate plasmid backbone fragments, a kanamycin selection gene and the chromoprotein gene, which differ in each assembly. First, PCR reagents for each respective fragment were mixed, and annealing temperature and extension times were calculated using the PCR optimization algorithm (Figure 2C). After thermal cycling, samples of each reaction were removed for gel electrophoresis. Amplified fragments of the correct lengths are shown in Figure 3A, where Fragments 2, 5 and 6 are backbone parts, and Fragments 1, 3, 4 and 7 are chromoprotein parts (Figure 3B).

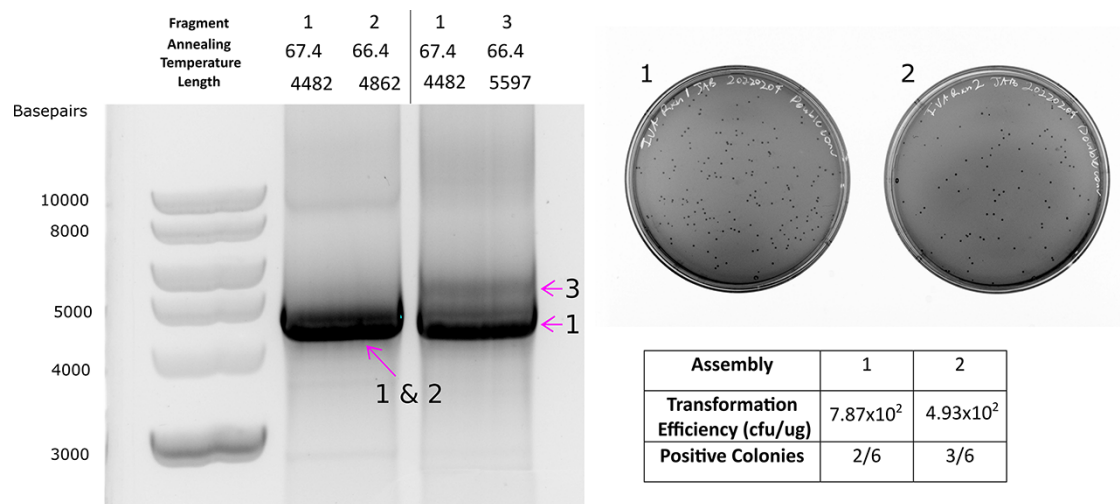
Following fragment amplification, the reactions were returned to the Opentrons robot, where AssemblyTron automatically performed a Dpn1 digestion to eliminate residual template DNA. Before proceeding to the assembly step, each fragment was cleaned and concentrated to remove polymerase, which we found to interfere with assembly. To do this, the protocol was paused, fragments were removed for cleaning and then they were returned to their positions on the thermocycler block. In future work, we plan to add an automated magnetic bead purification step. Next, a volume proportional to the fragment length for each resulting fragment was added to the respective assembly reaction in a new well. Again, for simplicity, we assumed that each fragment PCR is similarly efficient, and so this volume proportional to length will result in a roughly equimolar mixture of fragments. This avoids the need for further purification and quantification of DNA in most cases. Following the assembly, constructs were transformed into competent *E. coli* (produced in-house with a transformation efficiency  $>10^8$  CFU/ $\mu$ g pUC19) (25). Between 300 and 800 colonies were formed from each assembly with transformation efficiencies  $\sim 10^4$  CFU/ $\mu$ g of total assembly reaction DNA transformed. Greater than 98% of colonies were positive based

on chromoprotein expression. The Golden Gate assembly feature also includes the option to use a destination plasmid that contains internal BsaI sites specifically for Golden Gate cloning as opposed to assembling the plasmid backbone from PCR-amplified fragments.

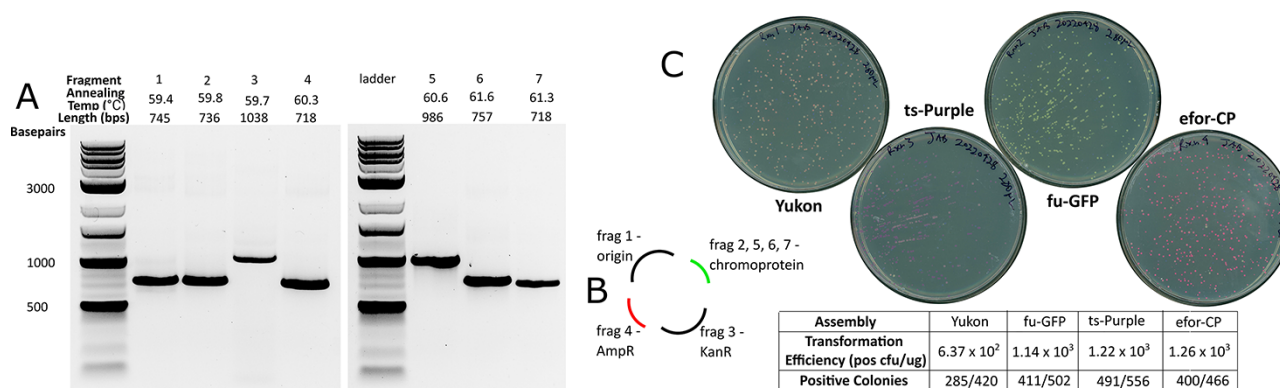
### 3.3 Short homology-based assembly

Since Gibson assembly is the most popular cloning strategy (27), we provide AssemblyTron users with a script to accommodate this approach. However, instead of creating a Gibson protocol with a lengthy assembly cycle requiring more reagents and time, we use a homology-dependent assembly strategy inspired by the IVA technique (7) and AQUA cloning (6). Similar to Gibson, IVA and AQUA cloning are based on recombination 20+ bps of homology between the ends of adjacent fragments. However, these strategies do not depend on enzyme-catalyzed creation of sticky ends, instead relying on assembly via homologous recombination *in vivo*, in *E. coli*. While we have successfully assembled two-fragment constructs in one-pot PCRs, as specified in IVA cloning (7), we experienced consistent failure when attempting to assemble constructs with more than two fragments in one pot. Perhaps, setting tighter tolerances for primer annealing temperature in the initial j5 design would improve one-pot fragment amplification. In order to ensure reproducibility and consistent success, we also developed a script to generate protocols using separate PCRs for each fragment as in AQUA cloning (6). The protocol for one-pot IVA reactions is still available in AssemblyTron, but we recommend a separate PCR script for more complex designs. In these protocols, following thermal cycling, PCRs, either combined or separate, are digested with Dpn1. For separate fragment PCRs, the products are then combined using volumes proportional to fragment length. This fragment mixture is then transformed into *E. coli*, where native *E. coli* machinery is responsible for assembling fragments.

To demonstrate short homology-based assembly with AssemblyTron, we performed two sets of combinatorial assemblies. First, we did a one-pot IVA to implement site-directed mutagenesis on a construct frequently used in the Wright Plant Synthetic Biology Lab at Virginia Tech. After initial PCR amplification of fragments and Dpn1 digestion to eliminate the residual template, we analyzed a small fraction of these reactions by gel electrophoresis to confirm correct fragment sizes (Figure 4A). Since both fragments in Lane 1 are approximately the same size,



**Figure 4.** Homology-dependent one-pot IVA with AssemblyTron. (A) AssemblyTron one-pot IVA was verified by amplification of appropriate bands via gel electrophoresis. Fragments 1 and 2 are indistinguishable due to size similarity; however there is a slight resolution between Fragments 1 and 3. (B) Consistent transformation efficiency and sequence verification validate the one-pot IVA protocol.



**Figure 5.** Homology-dependent AQUA assembly with AssemblyTron. (A) AssemblyTron AQUA protocol was verified by successful amplification of each chromoprotein assembly fragment. (B) Schematic to specify how fragments are assembled to yield final constructs. (C) Consistent transformation efficiency, correct antibiotic resistance and correct chromoprotein expression further validate the AQUA protocol.

they are indistinguishable. However, fragment sizes in Lane 2 differed by  $\sim 1000$  bp, and a slight separation between bands is seen in Figure 4A, indicated by barbed arrows and the fragment number. After amplification, Dpn1 digestion and column purification, we transformed these fragment mixtures into *E. coli*. We recorded transformation efficiencies  $\sim 10^2$  CFU/ $\mu$ g for these IVA reactions (Figure 4B). Sequence verification determined that out of six colonies, two contained the correct mutation for Assembly 1 and three were correct for Assembly 2. Incorrect colonies either had the original sequence or random insertions and deletions (Files S5 and S6).

In addition, we performed the same chromoprotein assembly described in Figure 3 using the AQUA assembly strategy. Correct bands were amplified (Figure 5A), fragment mixes were transformed into *E. coli* and dilutions were plated to determine transformation efficiency. We recorded transformation efficiencies  $\sim 10^3$  CFU/ $\mu$ g for these IVAs (Figure 5C). Recombination-based assembly techniques are characteristically lower in efficiency than restriction-ligation-based assemblies like Golden Gate, so these results were expected (28). Chromoprotein expression revealed that 68.7% of colonies in the Yukon assembly were positive for the Yukon chromoprotein and  $>80\%$  of colonies were positive for

the remaining colors. Negative colonies expressed either the blue chromoprotein from the template backbone vector or no color.

## 4. Discussion

The error-prone nature of molecular cloning inhibits the training of a large, robust workforce of synthetic biologists. AssemblyTron provides an open-source framework for automating cloning and is a tool to improve the productivity of laboratories plagued by failing workflows and limited funds (23). Its simple interface can be operated by an undergraduate student with minimal prior cloning experience. We demonstrate that our software is capable of seamlessly integrating with j5 DNA assembly designs. Thus, paired with j5, AssemblyTron provides reproducible algorithmic solutions to the design and build steps of the DBTL cycle ubiquitous to synthetic biology. AssemblyTron's ability to perform optimized PCRs and efficient DNA assemblies will allow researchers to spend more time asking questions and learning from experiments instead of scrutinizing error-prone cloning. AssemblyTron will also lighten the training barrier for undergraduates since all calculations and nuanced details are automatically handled by AssemblyTron.

Our data provide evidence that AssemblyTron delivers robust results for cloning workflows. We successfully performed PCRs with variable parameters and multipart assemblies via both IVA and Golden Gate. Assemblies performed with AssemblyTron in the OT-2 have similar transformation efficiency and accuracy to the manual procedures.

AssemblyTron is open source, under the Apache License 2.0, so any public laboratory interested in automating their workflow has access. The OT-2 liquid handling robot is one of the most affordable and can currently be purchased, complete with temperature module, magnetic module and thermocycler module, for \$16 500. This price point is 10 times less expensive than similar systems (29). By reducing opportunities for human error, decreasing human labor time and increasing throughput, automated cloning with AssemblyTron and the OT-2 can realistically save time and money in the long run.

Although AssemblyTron has expedited cloning in our laboratory and has the potential to do so in other laboratories as well, there are still numerous opportunities for improvement and expansion. The complexity of assemblies that can currently be handled by the AssemblyTron software is limited by our 96 combined primers and template capacity. However, as demand for multiplex assemblies increases, we will scale our platform to accommodate 384-well plates for the primer and template dilution steps. This modification will be simple to implement and serves as an example of the scalability of AssemblyTron.

Another drawback is that in the Golden Gate protocol, amplified fragments must be cleaned and concentrated before mixing with restriction enzymes and ligase, to prevent sticky ends from being filled in by residual polymerase. In the future, we will integrate the magnetic module to perform DNA cleaning within the OT-2 to avoid the use of expensive, time-consuming and human error-prone manual purification kits. Several open-source protocols for magnetic bead purification already exist for the OT-2 (30), which can be integrated into future versions of AssemblyTron. To increase the accessibility and interoperability of AssemblyTron, we hope to contribute our work to SynBioPython, a hardware automation Python library designed for biofoundry standardization (31).

Additionally, we are in the early stages of integrating a transformation protocol into AssemblyTron, which has also been achieved in an automated context in the past by DNA-BOT (21). This will save even more time for researchers if transformations can be performed directly in the OT-2. The more we can automate, the more time we will have to develop the field of synthetic biology.

## Supplementary data

Supplementary data are available at SYN BIO Online.

## Data availability

The authors confirm that the data supporting the findings of this study are available within the article and its supplementary materials.

## Material availability statement

The software, AssemblyTron, used to produce the results of this study is openly available on Zenodo at <http://doi.org/10.5281/zenodo.7250273>. The latest iteration of AssemblyTron is available on PyPI at <https://pypi.org/project/AssemblyTron/>.

## Funding

Research in the Wright Plant Synthetic Biology Laboratory: the United States Department of Agriculture National Institute of Food and Agriculture Agriculture and Food Research Initiative Plant Breeding for Agricultural Production Grant No. 2022-67013-36293 and Hatch Project [VA-1021738]; Virginia Space Grant Consortium; Virginia Tech Institute for Critical Technologies and Sciences Junior Faculty Award; and Virginia Tech College of Agriculture and Life Sciences Strategic Plan Advancement 2021 Integrated Internal Competitive Grants through the Center for Advanced Innovation in Agriculture.

## Acknowledgments

We thank Jennifer L. Nemhauser and Sebastian Cocioba for providing the plasmids used in this work. We also thank Patarasuda Chaisupa for designing the AssemblyTron Logo.

## Author contributions

J.A.B. and R.C.W. conceived of the project and designed experiments. All authors wrote code and revised code for the software package. J.A.B. acquired data. J.A.B. and R.C.W. analyzed and interpreted data. J.A.B. and R.C.W. drafted and revised the manuscript. All authors approved of the final manuscript.

*Conflict of interest statement.* The authors declare no conflicts of interest.

## References

- Casini, A., Storch, M., Baldwin, G.S. and Ellis, T. (2015) Bricks and blueprints: methods and standards for DNA assembly. *Nat. Rev. Mol. Cell Biol.*, **16**, 568–576.
- Appleton, E., Densmore, D., Madsen, C. and Roehner, N. (2017) Needs and opportunities in bio-design automation: four areas for focus. *Curr. Opin. Chem. Biol.*, **40**, 111–118.
- Li, M.Z. and Elledge, S.J. (2012) SLIC: a method for sequence- and ligation-independent cloning. In: Peccoud J. (ed). *Gene Synthesis. Methods in Molecular Biology*, Vol. 852. Humana Press, Totowa, NJ, pp. 51–59.
- Gibson, D.G. (2009) Synthesis of DNA fragments in yeast by one-step assembly of overlapping oligonucleotides. *Nucleic Acids Res.*, **37**, 6984–6990.
- Quan, J., Tian, J. and Ho, P.L. (2009) Circular polymerase extension cloning of complex gene libraries and pathways. *PLoS ONE*, **4**, e6441.
- Beyer, H.M., Gonschorek, P., Samodelov, S.L., Meier, M., Weber, W., Zurbruggen, M.D. and Chatterji, D. (2015) AQUA cloning: a versatile and simple enzyme-free cloning approach. *PLoS ONE*, **10**, e0137652.
- García-Nafria, J., Watson, J.F. and Greger, I.H. (2016) IVA cloning: a single-tube universal cloning system exploiting bacterial *in vivo* assembly. *Sci. Rep.*, **6**, 27459.
- Engler, C., Gruetzner, R., Kandzia, R., Marillonnet, S. and Peccoud, J. (2009) Golden Gate shuffling: a one-pot DNA shuffling method based on type IIS restriction enzymes. *PLoS ONE*, **4**, e5553.
- Kosuri, S. and Church, G.M. (2014) Large-scale *de novo* DNA synthesis: technologies and applications. *Nat. Methods*, **11**, 499–507.
- Palluk, S., Arlow, D.H., de Rond, T., Barthel, S., Kang, J.S., Bector, R., Baghdassarian, H.M., Truong, A.N., Kim, P.W., Singh, A.K. et al. (2018) *De novo* DNA synthesis using polymerase-nucleotide conjugates. *Nat. Biotechnol.*, **36**, 645–650.



11. Chao,R., Liang,J., Tasan,I., Si,T., Ju,L. and Zhao,H. (2017) Fully automated one-step synthesis of single-transcript TALEN pairs using a biological foundry. *ACS Synth. Biol.*, **6**, 678–685.
12. Edinburgh Genome Foundry. The University of Edinburgh. <https://www.ed.ac.uk/biology/research/facilities/edinburgh-genome-foundry> (3 June 2022, date last accessed).
13. Hillson,N., Caddick,M., Cai,Y., Carrasco,J.A., Chang,M.W., Curach,N.C., Bell,D.J., Le Feuvre,R., Friedman,D.C., Fu,X. et al. (2019) Building a global alliance of biofoundries. *Nat. Commun.*, **10**, 2040.
14. Vrana,J., de Lange,O., Yang,Y., Newman,G., Saleem,A., Miller,A., Cordray,C., Halabiya,S., Parks,M., Lopez,E. et al. (2021) Aquarium: open-source laboratory software for design, execution and data management. *Synth. Biol.*, **6**, ysab006.
15. Walsh,D.I., Pavan,M., Ortiz,L., Wick,S., Bobrow,J., Guido,N.J., Leinicke,S., Fu,D., Pandit,S., Qin,L. et al. (2019) Standardizing automated DNA assembly: best practices, metrics, and protocols using robots. *SLAS Technol.*, **24**, 282–290.
16. Hérisson,J., Duigou,T., du Lac,M., Bazi-Kabbaj,K., Sabeti Azad,M., Buldum,G., Telle,O., El Moubayed,Y., Carbonell,P., Swainston,N. et al. (2022) The automated Galaxy-SynBioCAD pipeline for synthetic biology design and engineering. *Nat. Commun.*, **13**, 5082.
17. Carbonell,P., Le Feuvre,R., Takano,E. and Scrutton,N.S. (2020) In silico design and automated learning to boost next-generation smart biomanufacturing. *Synth. Biol.*, **5**, ysaa020.
18. Tegally,H., San,J.E., Giandhari,J. and de Oliveira,T. (2020) Unlocking the efficiency of genomics laboratories with robotic liquid-handling. *BMC Genomics*, **21**, 729.
19. Hillson,N.J., Rosengarten,R.D. and Keasling,J.D. (2012) j5 DNA assembly design automation software. *ACS Synth. Biol.*, **1**, 14–21.
20. Jones,T.S., Oliveira,S.M.D., Myers,C.J., Voigt,C.A. and Densmore,D. (2022) Genetic circuit design automation with Cello 2.0. *Nat. Protoc.*, **17**, 1097–1113.
21. Storch,M., Haines,M.C. and Baldwin,G.S. (2020) DNA-BOT: a low-cost, automated DNA assembly platform for synthetic biology. *Synth. Biol.*, **5**, ysaa010.
22. Storch,M., Casini,A., Mackrow,B., Fleming,T., Trewhitt,H., Ellis,T., Baldwin,G.S. et al. (2015) BASIC: a new Biopart Assembly Standard for Idempotent Cloning provides accurate, single-tier DNA assembly for synthetic biology. *ACS Synth. Biol.*, **4**, 781–787.
23. Bryant,J., Kellinger,M., Longmire,C., Miller,R., Wright,R.C. et al. *PlantSynBioLab/AssemblyTron: AssemblyTron*. Zenodo; 2022. <https://doi.org/10.5281/zenodo.7250273> (25 October 2022, date last accessed).
24. Green,M.R., Sambrook,J. and Sambrook,J. (2012) *Molecular Cloning: A Laboratory Manual*, 4th edn. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, p. 3.
25. Hanahan,D., Jessee,J. and Bloom,F.R. (1991) Plasmid transformation of *Escherichia coli* and other bacteria. In: Miller J. H. (ed). *Bacterial Genetic Systems. Methods in Enzymology*, Vol. 204. Academic Press, Cambridge, MA, USA, pp. 63–113.
26. Davis,M.W. and Jorgensen,E.M. (2022) ApE, A plasmid Editor: a freely available DNA manipulation and visualization program. *Front. Bioinform.*, **2**.
27. Kahl,L.J. and Endy,D. (2013) A survey of enabling technologies in synthetic biology. *J. Biol. Eng.*, **7**, 13.
28. Pryor,J.M., Potapov,V., Pokhrel,N. and Lohman,G.J.S. (2022) Rapid 40 kb genome construction from 52 parts through data-optimized assembly design. *ACS Synth. Biol.*, **11**, 2036–2042.
29. Opentrons. *Open-source Lab Automation, Starting at \$5,000*. <https://opentrons.com/> (5 June 2022, date last accessed).
30. Magnetic Module GEN2. *Opentrons OT-2 Modular Hardware*. <https://opentrons.com/modules/magnetic-module/> (12 July 2022, date last accessed).
31. Yeoh,J.W., Swainston,N., Vegh,P., Zulkower,V., Carbonell,P., Holowko,M.B., Peddinti,G., Poh,C.L. et al. (2021) SynBiopython: an open-source software library for synthetic biology. *Synth. Biol.*, **6**, ysab001.