**Supporting Information**

# Expanding the limits towards 'one-pot' DNA assembly and transformation on a rapid-prototype microfluidic device

James M. Perry, [1, 3] Guy Soffer, [2, 3] Raja Jain,[2, 3] Steve C.C. Shih[1-3*]

[1]Department of Biology, Concordia University, 7141 Sherbrooke Street West, Montréal, Québec, Canada, H4B 1R6

[2]Department of Electrical and Computer Engineering, Concordia University, 1455 de Maisonneuve Blvd. West, Montréal, Québec, Canada, H3G 1M8

[3]Centre for Applied Synthetic Biology, Concordia University, 7141 Sherbrooke Street West, Montréal, Québec, Canada, H4B 1R6

[*]Corresponding author

Tel: +1-(514) 848-2424 x7579

Email: steve.shih@concordia.ca

## Table of contents

## Reagents and Materials

General chemicals were purchased from Sigma-Aldrich (St. Louis, MO). Miniprep kits (cat no. BS413) and gel extraction kits (cat no. BS354) were purchased from BioBasic (Amherst, NY). DMF fabrication materials include copper etchant (cat no. 415-4L) purchased from MG chemicals, single-sided printed circuit board substrates were purchased from PulsarProFX (Colorado Springs, CO), FluoroPel PFC1601 V from Cytonix LLC (Beltsville, MD), indium tin oxide (ITO)-coated glass slides, $R_S$ = 15–25 Ω (cat no. CG-61IN-S207, Delta Technologies, Loveland CO), Parafilm-M (cat no. PM998) from Bemis Company (Neenah, WI). Polydimethylsiloxane oil (cat no. DMS-T05) was purchased from Gelest (Morrisville, PA). Silica capillary tubing (cat no. 1068150381/10M) was purchased from Molex (Lisle, IL) and PEEK tubing (1/16" OD x .007" ID x 5 ft) was purchased from IDEX Health and Science (Wallingford, CT). For 3D printing, polylactic acid (PLA) filament was purchased from 3Dshop (Mississauga, ON). Type 44 Heat Sink Compound (cat no. 10-8120) used for connecting components of the temperature control system was purchased from GC Electronics (Rockford, IL).

## Strains, plasmids and media

Plasmids for this work are listed in **Table S6**. pFab plasmids was a gift from JBEI. pETM6-vioABECD[1] was a gift from Mattheos Koffas (Addgene plasmid # 66535 ; http://n2t.net/addgene:66535 ; RRID:Addgene_66535). DH5α (cat No. 18265017) purchased from Invitrogen (Waltham, MA) was used for plasmid storage at -80 °C. The BL21(DE3) strain (cat no. C600003) is from Invitrogen (Waltham, MA). SIG10 (cat no. CMC0001) cells (DH5α and DH10b derived) purchased from Sigma-Aldrich were used to prepare chemically competent

cells (protocol adapted from Hanahan et al. 1983 as written by Green el al. 2014[2,3]) for plasmid transformation both on and off-chip. For plasmid cultivation, *E. coli* strains were cultured in LB Miller broth and supplemented with 50 μg/mL kanamycin (kan) or 100 μg/mL ampicillin (amp). Plates for BL21(DE3) expression was additionally supplemented with 1 mM IPTG. For transformation recovery, super optimal broth with catabolite suppression (SOC) was used. All oligonucleotides used for PCR were purchased from BioCorp (Montreal, QC) and are listed in **Table S5**.

**Temperature control system**

The temperature control system was replicated from previous work[4] with some modifications. The system consists of three core components: the temperature sensor, the TEC module and the PID hardware/software. For the temperature sensor, a 100 kΩ NTC thermistor (Abra Electronics cat. no. 3D-HE-001) was used. A 20 mm x 40 mm x 6 mm aluminum heat-block was machined with a 1 mm diameter, 20 mm deep hole to house the thermistor and provide a contact bridge between the TEC module and the device. The TEC module (Tetech cat no. TE-31-1.4-1.15) was mounted on a large heatsink and the temperature sensor block was mounted on the TEC module. The 3D-printed holder provided space for the TEC-block combination to rest below the DMF device. All thermal contact points were applied with thermal-transfer paste (Abra Electronics, cat. no. 10-8120, Montreal, QC). The PID hardware consisted of a H-bridge (Digikey, cat no. MC33886PVM, Winnipeg, MN) drawing 11 V and up to 3 A from a GW Instek GPS-4303 DC power supply (ITM Instruments Inc., Montreal. QC) and delivered to the TEC module. The motor driver direction and gain were controlled through Arduino Uno (Digikey, cat no. A000066,

Winnipeg, MN) via the digital output ports. An in-house Python software managing the system can be downloaded on the project bitbucket (see: bitbucket.org/shihmicrolab/james_m_perry_dmf). The binary thermistor output was calibrated and converted to degrees Celsius (°C) using the Steinhart-Hart formula with constants provided by the product datasheet.

**COMSOL Modeling and Validation**

A COMSOL model of a bench-top heat shock transformation protocol was built to validate benchtop experiments (COMSOL.mph models available on bitbucket). Using the "Heat Transfer" module (and geometry tools), we modeled a microcentrifuge tube containing 50 μL of water partially submerged in a water bath to determine the temperature profile (and the time constant) during a heat shock process. The tube material was set to polypropylene using COMSOL's built in materials library. Initial temperatures for all domains were set to 0 °C with a boundary conditions (surface of tube exposed to water) was set to 42 °C. Before execution, a fine mesh of the model was generated. The model run-time was set to 30 seconds and the time-dependant solver with convergence tolerance = $10^{-3}$ to generate a temperature vs. time plot.

The temperature in a droplet was also modeled using COMSOL. A 3D model of the setup sculpted using device dimensions and domain materials defined using the materials library or manually inputting material parameters such as heat capacity, thermal conductivity and density based on information provided by manufacturers (in the case of ITO glass and FR-4 substrate). We simulated the temperature response of a droplet on the device by applying real temperature data from experiment as a boundary condition at the interface between the temperature sensor

block and the DMF device. The COMSOL model setup with the device top-plate gap distance set to 90 μm. The same solver and tolerance were used for simulating the temperature profile in a droplet. All material and thermal parameters were shown in **Table S7.**

**Volume replenishment system**

*Volume detection.* To detect volume changes on the device, we used a modified impedance detection circuit as described by Shih et al.[5] As shown in **Figure 1B**, the circuit consisted of a diode and a RC filter to rectify AC signals measured from the ITO top plate between 0 to 5 VDC. Since impedance cannot be directly measured, voltage is measured considering Ohm's law, R = V / I. To measure voltage, an application of a 10 ms pulse of 150 $V_{RMS}$ at a frequency of 15 kHz was applied to an electrode the voltage was measured across a 10 kΩ resistor. A diode was used to half-wave rectify the signal and a resistor and capacitor in parallel was used to smooth the rectified ripple. The rectified output signal from the top plate was delivered as an analog input signal into an Arduino Uno and converted into a 10-bit value (i.e. 0 - 1023) which we refer to as "volume bit value". Standard curves denoting these volume bit values at different driving potentials (150, 225, 300 $V_{RMS}$) were performed and were correlated to different droplet volumes ranging from 300 to 1300 nL (**Figure S10**).

By assuming that droplet height is constant (since the same double-sided tape is used in device preparation), a droplet's volume is a direct function of its surface area in contact with the top and bottom plates. Using the resistivity equation below, it is expected that resistance between an electrode and the top plate decreases as droplet volume increases. Using the capacitance

equation below, it is expected that capacitance between an electrode and the top plate increases as the droplet volume increases. Combining both formulas shows that impedance will decrease with increasing droplet volume.

$$\boldsymbol{R = \frac{\rho \cdot L}{A}}$$

*R*: Resistance
*ρ*: Surface area
*L*: length
*A*: Area

$$\boldsymbol{C = \varepsilon \frac{A}{d}}$$

*C*: capacitance
*ε*: Dielectric constant
*A*: Area
*d*: Plate distance

$$\boldsymbol{Z_{res} = R}$$
$$\boldsymbol{Z_{cap} = \frac{1}{j\omega C}}$$

*Z*: Impedance
*j*: Imaginary unit
*ω*: angular velocity

$$\boldsymbol{\frac{1}{Z_{eq}} = \frac{1}{Z_{res}} + \frac{1}{Z_{cap}}}$$

*Syringe pump*. A 3D printed syringe pump was designed using Autodesk Fusion 360. The design was based on a linear actuator designed by 3D Printing World hosted on thingaverse.com (thing: 2812734). One of the actuator's carrier cap and end piece were modified with notches to fit a gastight glass 500 µL syringes purchased from Hamilton (Reno, NV). The syringe was fitted with 300 mm of tubing (OD: 360 µm, ID: 150 µm) (Peek tubing, cat no. 1572, SciPro, Oshawa, ON). The end of the tube was fitted with 10 mm of glass capillary tubing (OD: 90 µm, ID: 21.5 µm) (Molex capillary tubing, cat no. 50-110-8574, Fisher scientific, Waltham, MA), and secured with Lepage Ultra Gel super glue (Henkel, Mississauga, ON). The capillary tubing was fitted between the top-plate and a reservoir electrode. The syringe pump stepper-motor was controlled using an in-house Python stepper-motor library (bitbucket.org/shihmicrolab/james_m._perry_dmf). The syringe was filled with up to 2.5 mL of de-ionized water. To regulate syringe pump operation, a PID algorithm was implemented using the volume bit value measured from the circuit.

*Operation* An in-house Python software (provided in the bitbucket repository) was written to implement the volume replenishment system. In the volume replenishment system, the droplet in the reservoir was sensed (using the protocol from *volume detection*) and translated to a volume bit

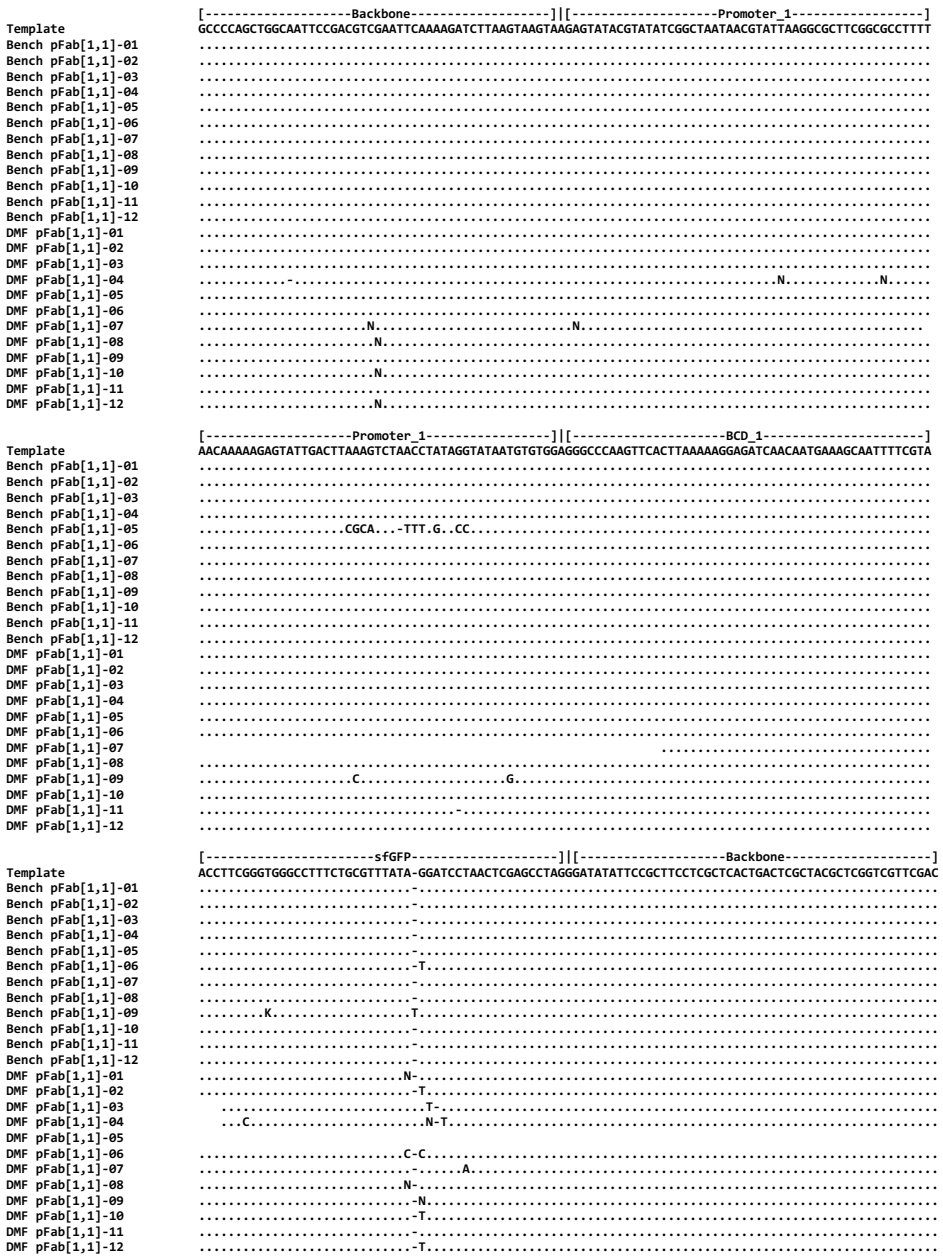
value ($Vol_{bit}$). The program periodically collected the bit data on a given electrode and decides whether to replenish the droplet with water (depending on the setpoint). The setpoint ($Vol_{set}$) was empirically determined by measuring the initial bit value of a droplet containing mixed assembly samples at an incubation region electrode. Typically, a 1250 nL droplet on an incubation electrode has bit values of between 100 and 200 bits depending on the thickness of parafilm and this value was stored as a setpoint variable, sample.setPoint. When sample.elecVol was greater than sample.setPoint, the volume of the sample had not evaporated, and no dispensing action had occurred. In contrast, when sample.elecVol was less than sample.setPoint, a droplet from a water reservoir was dispensed and actuated to the incubation electrode. A simulation model of the system was built using Python (in Bitbucket) to test the operation of the replenishment control system **Figure S11**. This consisted of a droplet generator method and an evaporation method to simulate the conditions of live incubation and replenishment scenarios. These methods had built-in variation and periodic error to aid in a model-based design of a robust controller.

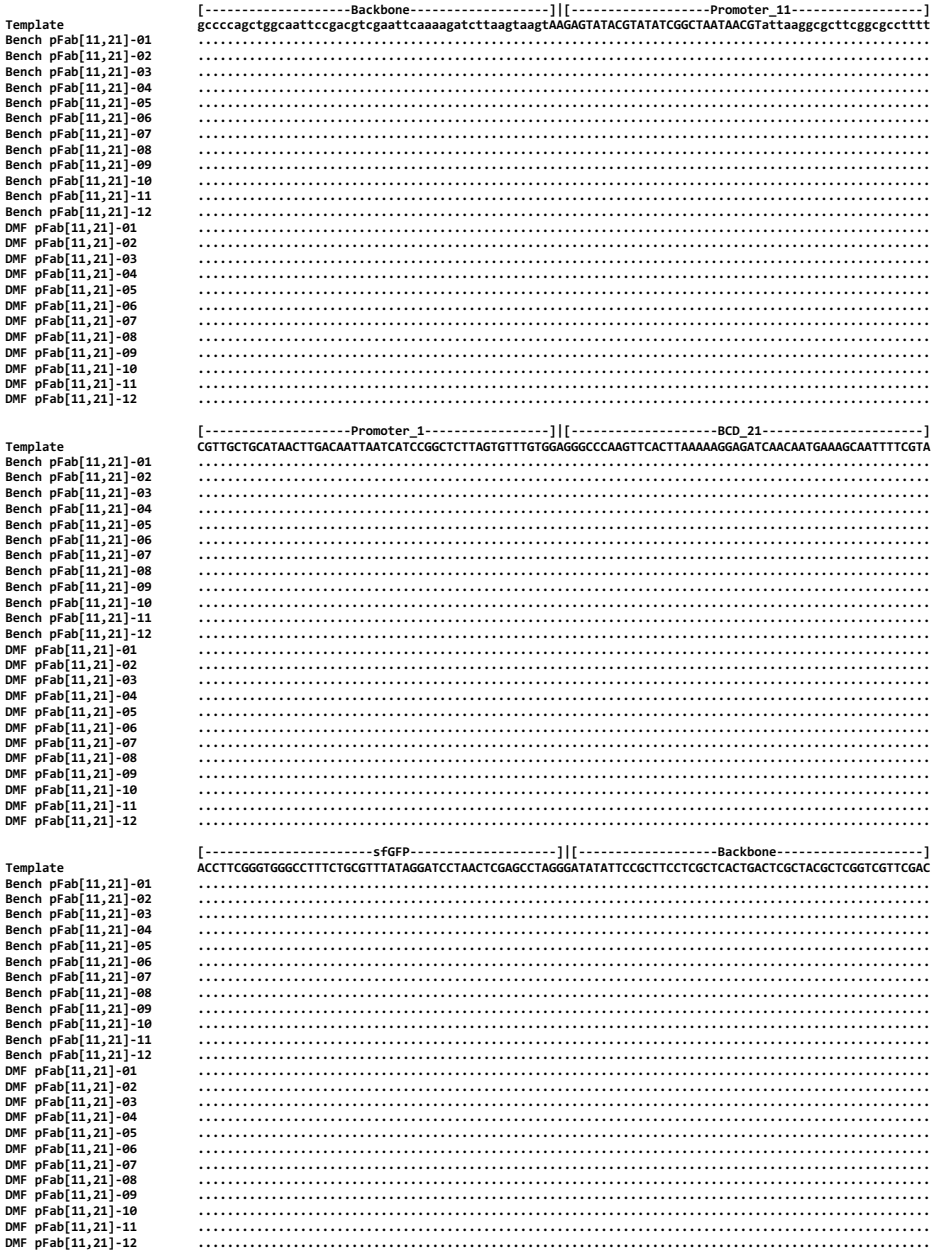
When establishing a set order for addressing samples for volume replenishment, we observed that some samples evaporated faster than others. This commonly resulted in some samples evaporating far below their desired setpoints while waiting for their turn. To address this, we established a system which prioritizes addressing samples requiring most urgent replenishment (i.e. sample.elecVol << sample.setPoint) - known as a "priority queue". The priority queue was designed to survey the sample volume bit values and ranks in descending priority. In design, when the priority queue is completed, a new queue is generated. When the syringe pumping action was triggered, the algorithm uses a PID algorithm to manage how many turns of the stepper motor is sufficient to deliver to the liquid to fill the reservoir. A 2 mm x 6 mm wick cut from a KimWipe was inserted under the ITO at a reservoir across from the syringe-fed reservoir to wick-away excess

water caused by the overshoot of the PID controller. This is managed by trigging droplet dispensing from the water reservoir electrode to the waste electrode when the volume bit value reaches over 15% of the setpoint. Additionally, this action is only taken if samples are adequately replenished. A detailed look of this volume replenishment system can be seen in **Figure 2C** and **Video S1**.

## Sequencing results

Grey-highlighted regions – 4 bp Eco31I-generated assembly junction



**Equation S1**

Rise: $\mathrm{T}(t) = \left(1 - e^{\frac{-t}{\tau}}\right)(\mathrm{T}_{max} - \mathrm{T}_{min}) + \mathrm{T}_{min}$

**Equation S2**

Fall: $\mathrm{T}(t) = \left(e^{\frac{-t}{\tau}}\right)(\mathrm{T}_{max} - \mathrm{T}_{min}) + \mathrm{T}_{min}$

T = temperature (°C)
$t$ = time (s)
$\tau$ = time constant (s)

**Table S1. 50 µL PCR reaction**

| **Component (added in order)** | **Volume (µL)** |
|---|---|
| DI water | to 50 |
| 5X Phusion HF Buffer | 10 |
| Forward Primer (10 µM) | 2.5 |
| Reverse Primer (10 µM) | 2.5 |
| *Template (1-10 ng) | 1 |
| Phusion | 0.5 |

**omitted for overlapp extenstion PCR*

**Table S2. PCR thermal cycling conditions**

| **Step** | | **Temp (°C)** | **Time** |
|---|---|---|---|
| Initial Denaturation | | 98 | 3 min |
| 35 Cycles | Denaturation | 98 | 30 sec |
| | Annealing | 65 | 30 sec |
| | Extension | 72 | 30 sec/kb |
| Final Extension | | 72 | 10 min |
| Hold | | 4 | |

**Table S3. Golden Gate thermal cycling conditions**

| **step** | | **temp (°C)** | **time** |
|---|---|---|---|
| initial Digestion | | 37 | 10 min |
| 5,15 or 25 Cycles | digestion | 37 | 1.5 min |
| | ligation | 16 | 3 min |
| final digestion | | 37 | 10 min |

**Table S4. Benchtop control reactions**

| Experiment | Plasmid | Assembly | Transformation | Figure |
|---|---|---|---|---|
| Water Replenishment | pFAB | Parts/thermocycler | Water bath and ice | Figure 2C |
| Heat-shock profiling | pUC19 | Full plasmid (already assembled) | Water bath and ice | Figure 3B/3C |
| Assembly/Transformation on device | pFAB/pVio | Parts/thermocycler | Water bath and ice | Figure 4B |

**Table S5: Primers used in this study**

Legend: Barcode, T7 mutation, BsaI/Eco31I site

**Primers for pFAB plasmid assembly**

| No. | Name | 5'-3' sequence | Template |
|---|---|---|---|
| RP_1 | RP_Prom_F | cacaccaGGTCTCCagagtatacgtatatcggctaataacgtattaa | universal forward |
| RP_2 | RP_Prom-1_R | cacaccaGGTCTCCctccacacattatacctataggttagacttta | pFAB4876 |
| RP_3 | RP_Prom-2_R | cacaccaGGTCTCCctccacacattataggtacaaaaagatg | pFAB4884 |
| RP_4 | RP_Prom-9_R | cacaccaGGTCTCCctccacacattatacgagccg | pFAB4924 |
| RP_5 | RP_Prom-11_R | cacaccaGGTCTCCctccacaaacactaagagccg | pFAB4932 |
| RP_6 | RP_BCD-gfp_F | cacaccaGGTCTCCggagggcccaagttcactta | universal for *BCD-gfp* varients on: pFAB4876, pFAB4877, pFAB4882, pFAB4883 |
| RP_7 | RP_BCD-gfp_R | cacaccaGGTCTCCgatacctaggctcgagttagga | |
| RP_8 | RP_BACKBONE_F | cacaccaGGTCTCCgatatattccgcttcctcgctca | univeral for pFAB backbone |
| RP_9 | RP_BACKBONE_R | cacaccaGGTCTCCctcttacttacttaagatctttgaattcgac | |

**Primers for T7 promoter synthesis - VJ(s)-T7_promoter**

| No. | Name | 5'-3' sequence | Template |
|---|---|---|---|
| VJ_55 | VJ(s)_C4_F | CAGTTCATGTATTGGGAGAGctaggatcgagatcgatctcgatcccgcgaaattaatacgactcactaTCAAGgaattgtgagcg | primer pair annealing |
| VJ_56 | VJ(s)_C4_R | atgtatatctccttcttaaagttaaacaaaattatttctagaggggaattgttatccgctcacaattcCTTGAtagtgag | |
| VJ_57 | VJ(s)_Cons_F | TATCCTCCACAAGATCTAGTctaggatcgagatcgatctcgatcccgcgaaattaatacgactcactaTAGGGgaattgtga | |
| VJ_58 | VJ(s)_Cons_R | atgtatatctccttcttaaagttaaacaaaattatttctagaggggaattgttatccgctcacaattcCCCTAtagtgagtcg | |
| VJ_59 | VJ(s)_H10_F | GGTATGGTGGTATAGTAAGCctaggatcgagatcgatctcgatcccgcgaaattaatacgactcactaCGGAAgaattgtga | |
| VJ_60 | VJ(s)_H10_R | atgtatatctccttcttaaagttaaacaaaattatttctagaggggaattgttatccgctcacaattcTTCCGtagtgagtcg | |
| VJ_61 | VJ(s)_H9_F | TCGTACTGTGATACACGCGActaggatcgagatcgatctcgatcccgcgaaattaatacgactcactaATACTgaattgtgag | |
| VJ_62 | VJ(s)_H9_R | atgtatatctccttcttaaagttaaacaaaattatttctagaggggaattgttatccgctcacaattcAGTATtagtgagtcgtatt | |
| VJ_63 | VJ(s)_G6_F | CAGGGGTAGGACCATCAGTActaggatcgagatcgatctcgatcccgcgaaattaatacgactcactaTTTCGgaattgtg | |
| VJ_64 | VJ(s)_G6_R | atgtatatctccttcttaaagttaaacaaaattatttctagaggggaattgttatccgctcacaattcCGAAAtagtgagtcgtatt | |

**Level -1 Primers to assemble entry vector - pETVJ (entry vector)**

| No. | Name | 5'-3' sequence | Template |
|---|---|---|---|
| VJ_1 | VJ(-1)_KAN_F | cacaccaGGTCTCCTGtcagataaaatatttctagatttcagtgcaatttatc | pFAB4876 |
| VJ_2 | VJ(-1)_KAN_R | cacaccaGGTCTCCAAttacttacttaagatcttttgaattcgacgt | |
| VJ_3 | VJ(-1)_ORI_F | cacaccaGGTCTCCAAttgagatccttttttctgcgcg | pETM6 |
| VJ_4 | VJ(-1)_T7term_R | cacaccaGGTCTCCGAcaaaaaacccctcaagacccg | |

**Level 0 Primers to assemble expression vectors - pETVJ-Prom-Vio**

| No. | Name | 5'-3' sequence | Template |
|---|---|---|---|
| VJ_5 | VJ(0)_Vector_F | cacaccaGGTCTCCgagtctggtaaagaaaccgctg | pETVJ |
| VJ_6 | VJ(0)_Vector_R | cacaccaGGTCTCCgctacgccggacgc | |
| VJ_7 | VJ(0)_Prom_C4_F | cacaccaGGTCTCCTAGCCAGTTCATGTATTGGGAGAGctag | VJ(s)-C4 |
| VJ_8 | VJ(0)_Prom_Cons_F | cacaccaGGTCTCCTAGCTATCCTCCACAAGATCTAGTctaggat | VJ(s)-Cons |
| VJ_9 | VJ(0)_Prom_G6_F | cacaccaGGTCTCCTAGCCAGGGGTAGGACCATCAGTAc | VJ(s)-G6 |
| VJ_10 | VJ(0)_Prom_H9_F | cacaccaGGTCTCCTAGCTCGTACTGTGATACACGCGA | VJ(s)-H9 |
| VJ_11 | VJ(0)_Prom_H10_F | cacaccaGGTCTCCTAGCGGTATGGTGGTATAGTAAGCctagg | VJ(s)-H10 |
| VJ_12 | VJ(0)_Prom_R | cacaccaGGTCTCCatgtatatctccttcttaaagttaaacaaaattatttct | VJ(s)-(T7_promoter) |
| VJ_13 | VJ(0)_VIOA_F | cacaccaGGTCTCCACATatgttaaatttagagcattcagtttcaattgt | pETM6-ABECD |
| VJ_14 | VJ(0)_VIOA_R | cacaccaGGTCTCCACTCtcaaaggtatactacttctttcacagtttc | |
| VJ_15 | VJ(0)_VIOB_F | cacaccaGGTCTCCACATatgagtgttttagattttcctcgaataca | |
| VJ_16 | VJ(0)_VIOB_R | cacaccaGGTCTCCACTCctaaccttcctttgaaagagttccg | |
| VJ_17 | VJ(0)_VIOE_F | cacaccaGGTCTCCACATatggaattacgtaaagtagatagagttccg | |
| VJ_18 | VJ(0)_VIOE_R | cacaccaGGTCTCCACTCtcaattcctatgagagagacttgttacat | |
| VJ_19 | VJ(0)_VIOC_F | cacaccaGGTCTCCACATatgagtaaaataattattgttggtggtggg | |
| VJ_20 | VJ(0)_VIOC_R | cacaccaGGTCTCCACTCttaattcattctccctattttgtaccaaacat | |
| VJ_21 | VJ(0)_VIOD_F | cacaccaGGTCTCCACATatgaacattttagtgatcggggc | |
| VJ_22 | VJ(0)_VIOD_R | cacaccaGGTCTCCACTCgagttaacgttgcagcgca | |

**Level 1 primers to assemble vio pathway vectors - pETVJ-VioABECD**

| No. | Name | 5'-3' sequence | Template |
|---|---|---|---|
| VJ_23 | VJ(1)_Vector_F | cacaccaGGTCTCCtcagataaaatatttctagatttcagtgcaatttatc | pETVJ |
| VJ_24 | VJ(1)_Vector_R | cacaccaGGTCTCCgctacgccggacgc | |
| VJ_25 | VJ(1)_Prom_C4-A_F | cacaccaGGTCTCCTAGCCAGTTCATGTATTGGGAGAGctag | pETVJ-C4-VioA |
| VJ_26 | VJ(1)_Prom_Cons-A_F | cacaccaGGTCTCCTAGCTATCCTCCACAAGATCTAGTctaggat | pETVJ-Cons-VioA |

| | | | |
|---|---|---|---|
| VJ_27 | VJ(1)_Prom_G6-A_F | cacaccaGGTCTCCTAGCCAGGGGTAGGACCATCAGTAc | pETVJ-G6-VioA |
| VJ_28 | VJ(1)_Prom_H9-A_F | cacaccaGGTCTCCTAGCTCGTACTGTGATACACGCGA | pETVJ-H9-VioA |
| VJ_29 | VJ(1)_Prom_H10-A_F | cacaccaGGTCTCCTAGCGGTATGGTGGTATAGTAAGCctagg | pETVJ-H10-VioA |
| VJ_30 | VJ(1)_Prom_C4-B_F | cacaccaGGTCTCCTGCCCAGTTCATGTATTGGGAGAGctag | pETVJ-C4-VioB |
| VJ_31 | VJ(1)_Prom_Cons-B_F | cacaccaGGTCTCCTGCCTATCCTCCACAAGATCTAGTctaggat | pETVJ-Cons-VioA |
| VJ_32 | VJ(1)_Prom_G6-B_F | cacaccaGGTCTCCTGCCCAGGGGTAGGACCATCAGTAc | pETVJ-G6-VioB |
| VJ_33 | VJ(1)_Prom_H9-B_F | cacaccaGGTCTCCTGCCTCGTACTGTGATACACGCGA | pETVJ-H9-VioB |
| VJ_34 | VJ(1)_Prom_H10-B_F | cacaccaGGTCTCCTGCCGGTATGGTGGTATAGTAAGCctagg | pETVJ-H10-VioB |
| VJ_35 | VJ(1)_Prom_C4-E_F | cacaccaGGTCTCCGCAACAGTTCATGTATTGGGAGAGctag | pETVJ-C4-VioE |
| VJ_36 | VJ(1)_Prom_Cons-E_F | cacaccaGGTCTCCGCAATATCCTCCACAAGATCTAGTctaggat | pETVJ-Cons-VioE |
| VJ_37 | VJ(1)_Prom_G6-E_F | cacaccaGGTCTCCGCAACAGGGGTAGGACCATCAGTAc | pETVJ-G6-VioE |
| VJ_38 | VJ(1)_Prom_H9-E_F | cacaccaGGTCTCCGCAATCGTACTGTGATACACGCGA | pETVJ-H9-VioE |
| VJ_39 | VJ(1)_Prom_H10-E_F | cacaccaGGTCTCCGCAAGGTATGGTGGTATAGTAAGCctagg | pETVJ-H10-VioE |
| VJ_40 | VJ(1)_Prom_C4-C_F | cacaccaGGTCTCCACTACAGTTCATGTATTGGGAGAGctag | pETVJ-C4-VioC |
| VJ_41 | VJ(1)_Prom_Cons-C_F | cacaccaGGTCTCCACTATATCCTCCACAAGATCTAGTctaggat | pETVJ-Cons-VioC |
| VJ_42 | VJ(1)_Prom_G6-C_F | cacaccaGGTCTCCACTACAGGGGTAGGACCATCAGTAc | pETVJ-G6-VioC |
| VJ_43 | VJ(1)_Prom_H9-C_F | cacaccaGGTCTCCACTATCGTACTGTGATACACGCGA | pETVJ-H9-VioC |
| VJ_44 | VJ(1)_Prom_H10-C_F | cacaccaGGTCTCCACTAGGTATGGTGGTATAGTAAGCctagg | pETVJ-H10-VioC |
| VJ_45 | VJ(1)_Prom_C4-D_F | cacaccaGGTCTCCTTACCAGTTCATGTATTGGGAGAGctag | pETVJ-C4-VioD |
| VJ_46 | VJ(1)_Prom_Cons-D_F | cacaccaGGTCTCCTTACTATCCTCCACAAGATCTAGTctaggat | pETVJ-Cons-VioD |
| VJ_47 | VJ(1)_Prom_G6-D_F | cacaccaGGTCTCCTTACCAGGGGTAGGACCATCAGTAc | pETVJ-G6-VioD |
| VJ_48 | VJ(1)_Prom_H9-D_F | cacaccaGGTCTCCTTACTCGTACTGTGATACACGCGA | pETVJ-H9-VioD |
| VJ_49 | VJ(1)_Prom_H10-D_F | cacaccaGGTCTCCTTACGGTATGGTGGTATAGTAAGCctagg | pETVJ-H10-VioD |
| VJ_50 | VJ(1)_T7termA_R | cacaccaGGTCTCCGGCAcaaaaaacccctcaagacccg | pETVJ-(T7_promoter)-VioA |
| VJ_51 | VJ(1)_T7termB_R | cacaccaGGTCTCCTTGCcaaaaaacccctcaagacccg | pETVJ-(T7_promoter)-VioB |
| VJ_52 | VJ(1)_T7termE_R | cacaccaGGTCTCCTAGTcaaaaaacccctcaagacccg | pETVJ-(T7_promoter)-VioE |
| VJ_53 | VJ(1)_T7termC_R | cacaccaGGTCTCCGTAAcaaaaaacccctcaagacccg | pETVJ-(T7_promoter)-VioC |
| VJ_54 | VJ(1)_T7termD_R | cacaccaGGTCTCCCTGAcaaaaaacccctcaagacccg | pETVJ-(T7_promoter)-VioD |

**Colony PCR primers for pETVJ-VioABECD**

| No. | Name | 5'-3' sequence | Template |
|---|---|---|---|
| VJ_65 | pBRrevBAM_F | ggtgatgtcggcgatatagg | |
| VJ_66 | KanRcp_R | ctatcgccttcttgacgagttc | |
| VJ_67 | VJ(CP)_BB-A_F | ggcgatataggcgccagcaacc | |
| VJ_68 | VJ(CP)_BB-A_R | ttggatccgtccaccaacgcgg | pETVJ-Vio[4,4,4,4,4] and pETVJ-Vio[3,3,2,2,1] |

| | | |
|---|---|---|
| VJ_69 | VJ(CP)_A-B_F | agccagttgcctacacatcgtga |
| VJ_70 | VJ(CP)_A-B_R | tgggtatttcggtttgcggttgg |
| VJ_71 | VJ(CP)_B-E_F | ctgacctcgcagccggaactc |
| VJ_72 | VJ(CP)_B-E_R | ggacatccacaaacggtgtccca |
| VJ_73 | VJ(CP)_E-C_F | aatggtcacgggtgatccgaacg |
| VJ_74 | VJ(CP)_E-C_R | gaaacttcaacgccgcacgcc |
| VJ_75 | VJ(CP)_C-D_F | tgcaagatatggcgcgggcaaac |
| VJ_76 | VJ(CP)_C-D_R | gattttcgctcagcgccgcataag |
| VJ_77 | VJ(CP)_D-BB_F | gcagtggttggcgcacagttac |
| VJ_78 | VJ(CP)_D-BB_R | gacgaagagcatcaggggctcg |

**Table S6: Strains and plasmids used in this study**

| Strain | Genotype | Source or Reference |
|---|---|---|
| SIG10 | F- mcrA Δ(mrr-hsdRMS-mcrBC) endA1 recA1 Φ80dlacZΔM15 ΔlacX74 araD139 Δ(ara, leu)7697 galU galK rpsL nupG λ- tonA | Sigma aldrich |
| DH5α | F- Φ80*lac*ZΔM15 Δ(*lac*ZYA-*arg*F) U169 *rec*A1 *end*A1 *hsd*R17(rk-, mk+) *pho*A *sup*E44 *thi*-1 *gyr*A96 *rel*A1 λ- | ThermoFisher |

| Plasmid | Description | Source |
|---|---|---|
| pFAB 4876 | *Prom1-BCD1-gfp* | JBEI (JPUB_001384) |
| pFAB 4877 | *Prom1-BCD2-gfp* | JBEI (JPUB_001383) |
| pFAB4882 | *Prom1-BCD20-gfp* | JBEI (JPUB_001382) |
| pFAB4883 | *Prom1-BCD21-gfp* | JBEI (JPUB_001381) |
| pFAB4884 | *Prom2-BCD1-gfp* | JBEI (JPUB_001380) |
| pFAB4924 | *Prom9-BCD1-gfp* | JBEI (JPUB_001376) |
| pFAB4932 | *Prom11-BCD1-gfp* | JBEI (JPUB_001372) |
| pFAB[1,1] | *Prom1-BCD1-gfp* | This study |
| pFAB[11,21] | *Prom11-BCD21-gfp* | This study |
| pUC19 | | Thermofisher (prepared in-house) |
| pETM6-vioABECD | *cons-VioA - cons-VioB - cons-VioE - cons-VioC - cons-VioD* | Jones et al. (2015)[1] Addgene Plasmid #66535 |
| pETVJ | pETM6 with *KanR* replacing *AmpR* | This study |
| pETVJ-C4-VioA | *C4-VioA* | This study |
| pETVJ-Cons-VioA | *Cons-VioA* | This study |
| pETVJ-G6-VioA | *G6-VioA* | This study |
| pETVJ-H9-VioA | *H9-VioA* | This study |
| pETVJ-H10-VioA | *H10-VioA* | This study |
| pETVJ-C4-VioB | *C4-VioB* | This study |
| pETVJ-Cons-VioA | *Cons-VioA* | This study |
| pETVJ-G6-VioB | *G6-VioB* | This study |
| pETVJ-H9-VioB | *H9-VioB* | This study |
| pETVJ-H10-VioB | *H10-VioB* | This study |
| pETVJ-C4-VioE | *C4-VioE* | This study |
| pETVJ-Cons-VioE | *Cons-VioE* | This study |
| pETVJ-G6-VioE | *G6-VioE* | This study |
| pETVJ-H9-VioE | *H9-VioE* | This study |
| pETVJ-H10-VioE | *H10-VioE* | This study |
| pETVJ-C4-VioC | *C4-VioC* | This study |
| pETVJ-Cons-VioC | *Cons-VioC* | This study |
| pETVJ-G6-VioC | *G6-VioC* | This study |
| pETVJ-H9-VioC | *H9-VioC* | This study |
| pETVJ-H10-VioC | *H10-VioC* | This study |
| pETVJ-C4-VioD | *C4-VioD* | This study |
| pETVJ-Cons-VioD | *Cons-VioD* | This study |

| | | |
|---|---|---|
| pETVJ-G6-VioD | *G6-VioD* | This study |
| pETVJ-H9-VioD | *H9-VioD* | This study |
| pETVJ-H10-VioD | *H10-VioD* | This study |
| pETVJ-Vio[1,1,1,1,1] | *cons-VioA - cons-VioB - cons-VioE - cons-VioC - cons-VioD* | This study |
| pETVJ-Vio[3,3,2,1,1] | *H10-VioA - H10s-VioB - H9-VioE - G6-VioC - G6-VioD* | This study |

**Table S7. COMSOL material thermal paramters**

| **Material** | **Density, ρ (kg/m³)** | **Thermal conductivity, k W/(m·K)** | **Heat capacity at a constant pressure, $C_p$ J/(kg·K)** |
|---|---|---|---|
| Glass | 2210 | 1.4 | 730 |
| Aluminum | 2700 | 238 | 900 |
| Water | 838.466135+1.40050603 *T^1-0.0030112376 *T^2 +3.71822313E-7 *T^3 | -0.869083936 +0.00894880345 *T^1-1.58366345E-5 *T^2 +7.97543259E-9 *T^3 | 12010.1471-80.4072879 *T^1 +0.309866854 *T^2 -5.38186884E-4 *T^3 +3.62536437E-7 *T^4 |
| Air | Partial derivatives pA: d(pA*0.02897/8.314/T,pA) T: d(pA*0.02897/8.314/T,T) | -0.00227583562 +1.15480022E-4 *T^1 -7.90252856E-8 *T^2+4.11702505E-11 *T^3 -7.43864331E-15 *T^4 | 1047.63657 -0.372589265 *T^1 +9.45304214E-4 *T^2 -6.02409443E-7 *T^3 +1.2858961E-10 *T^4 |

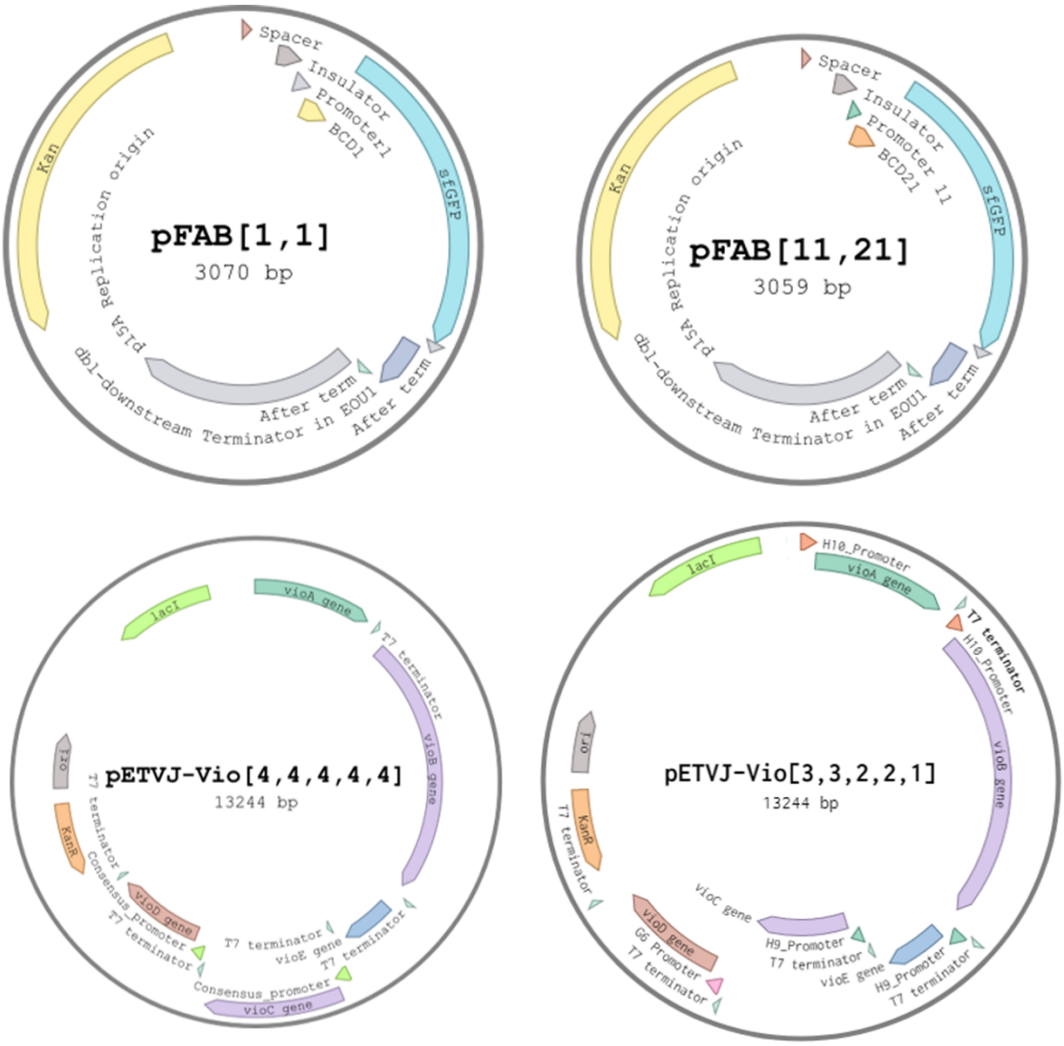



**Figure S1: Maps of plasmids assembled on device.** pFab plasmids represent 3-fragment assemblies comprised of a promoter, *BCD* element (synthetically engineered shine Dalgarno sequence) linked to *sfGFP* and the vector backbone including *p15A* origin of replication and *kanR* cassette. pFab[1,1] contains *promoter 1* and *BCD 1* and pFab[11,21] contain from *promoter 11* and *BCD 21* engineered by Mutalik et al. (2013)[6]. pETVJ-Vio plasmids contain the five gene cassettes comprising the violacein pathway (ordered: *VioA*, *VioB*, *VioE*, *VioC*, *VioD*) with each gene regulated by a *T7* promoter variant described by Jones et al. (2015)[1]. For simplicity, bracketed nomenclature refers to the strength of promoter for each cassette. 4=C4, 3=H10, 2=H9, 1=G6. The assembly represents 5 gene cassette fragments plus the vector backbone fragment including a *ColEI* origin of replication and a *kanR* cassette.

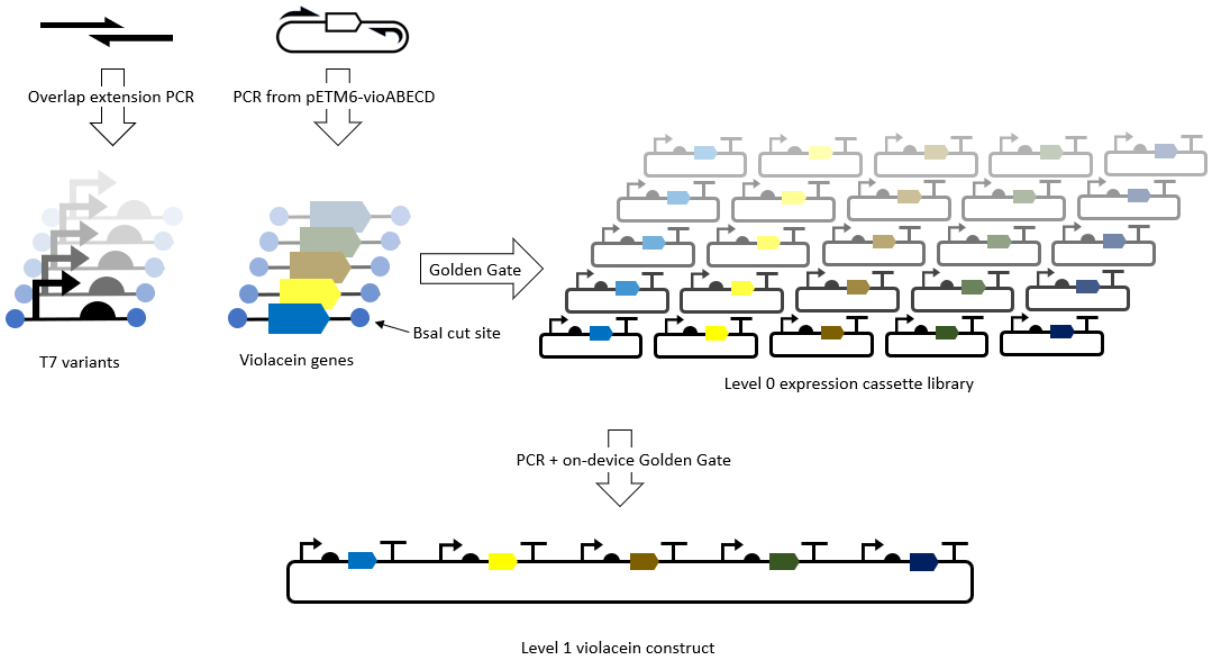



**Figure S2: The hierarchal assembly of pETVJ-Vio constructs.** Prior to on-device assembly, violacein gene cassettes were built. This involved first, creating the T7 promoter variants via overlap-extension PCR. Each promoter (*C4*, *H10*, *H9*, *G6*) was combined with each violacein gene *VioA*, *VioB*, *VioE*, *VioC*, *VioD* (amplified from pETM6-VIOABECD) into an entry vector containing a T7 terminator sequence, *ColEI* origin of replication and a *kanR* cassette. Selected cassettes were then amplified and combined into a violacein constructs containing all five genes. Assemblies were done via Golden Gate.

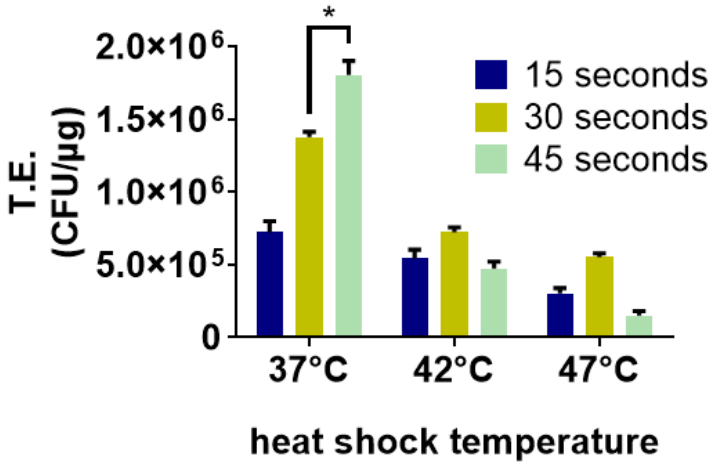



**Figure S3: Benchtop optimization of temperature conditions for heat shock transformation.** To begin to optimize heat shock conditions on-device, heat shock times and temperatures were compared using standard bench-scale conditions. Each transformation was performed in triplicate using 1.7 mL polypropylene microcentrifuge tubes containing 50 µL competent cells mixtures and 10 ng of mini-prepped pUC19 plasmid.

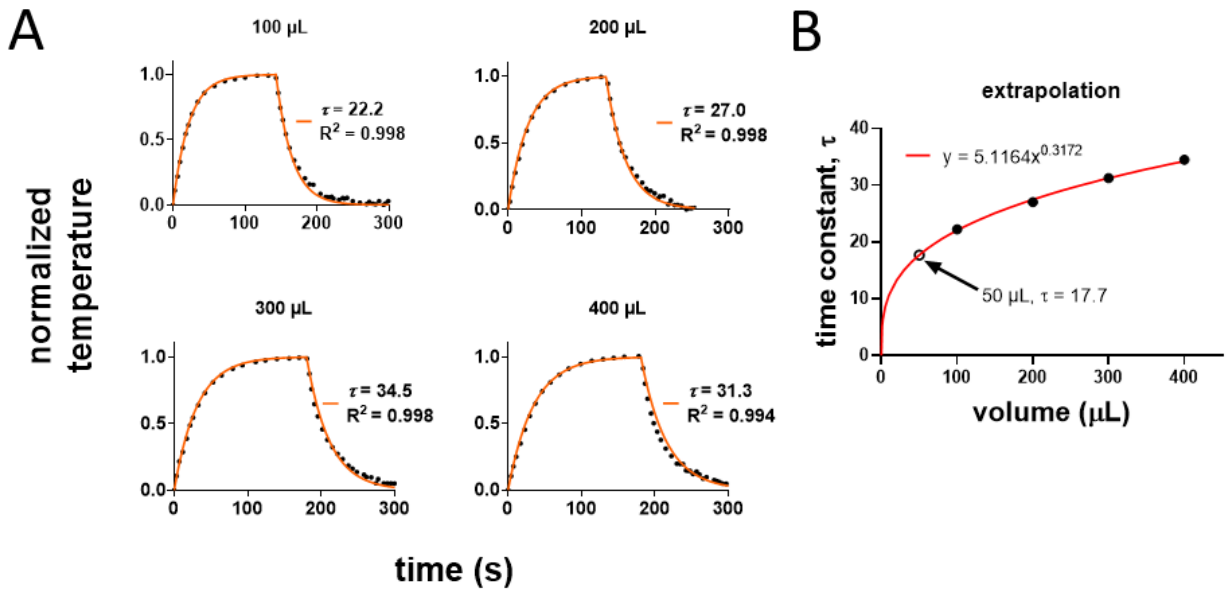



**Figure S4: Extrapolation of time constant data to determine time constant for 50 uL competent cell sample.** It was anticipated that measuring an accurate time constant for a heat shock profile of 50 uL cell sample would be difficult given the size of the temperature probe. To circumvent this challenge, a standard curve was made based on time constants from heat-shocking cell samples ranging from 100 to 400 uL. This was done by submerging a thermistor probe into an 1.7 mL microcentrifuge tube containing the different sample volumes and measuring the temperature profile of a heat shock from ice water to ~42 °C. Samples we kept in the warm water bath until stability at a maximum temperature was observed. Since precise max and min temperatures are not necessary between samples, the curves were normalized to a range between 0 and 1. A) Next, a time constant was selected to fit a rise and fall curve to the experimental data producing a maximum $R^2$ value. B) Using the generated analytical model, we extrapolated the time constant for a 50 uL sample temperature curve. Additionally, a COMSOL simulation of these conditions applied to a 50 uL water sample gave a similar time constant of 17.

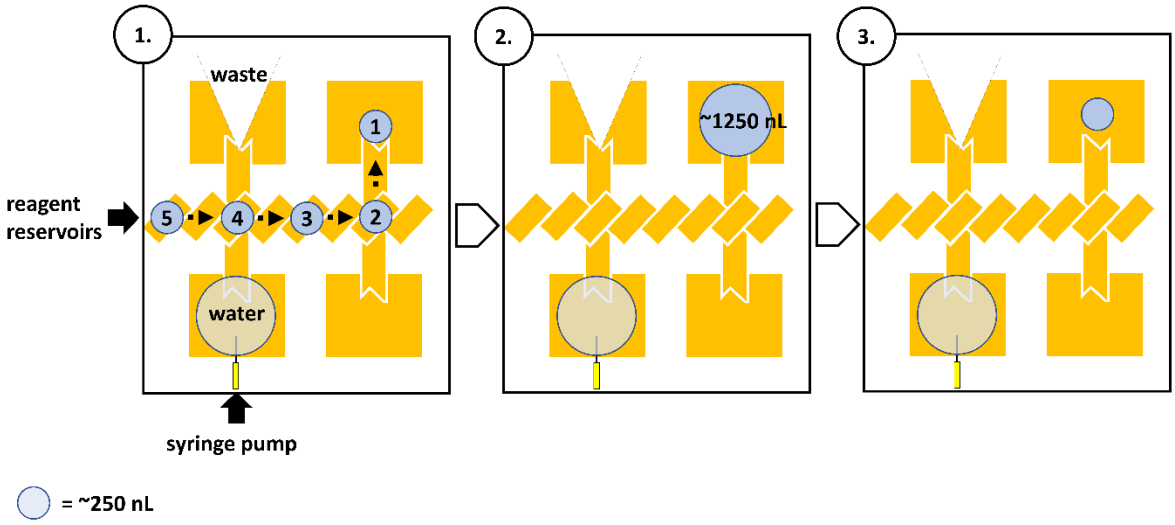



**Figure S5: Device workflow for 3-part assembly.** (1) Droplets containing DNA fragments (droplet 1-3: promoter, drop *BCD-gfp*, plasmid backbone) and assembly master mix (droplets 4 and 5) are dispensed and (2) the merged 1.25 µL droplet sent to an incubation electrode where they are mixed and thermocycled as per a standard MoClo protocol (**Table S3**). (3) Upon completion of thermocycling, the droplet is left to evaporate, removing water volume to avoid diluting the competent cells.

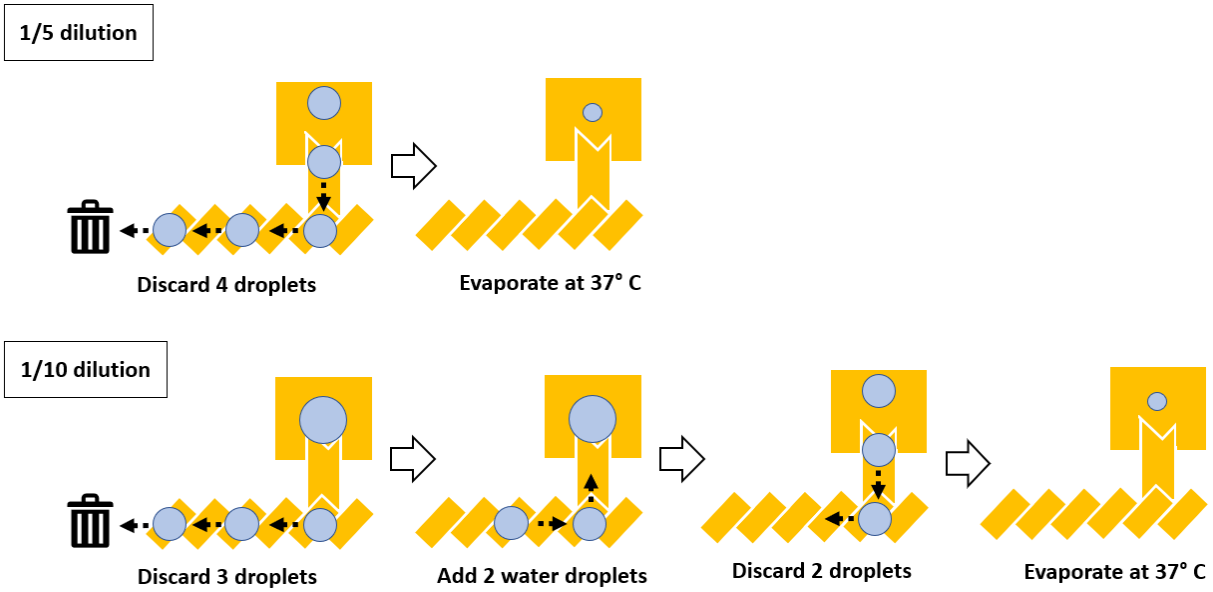



**Figure S6: Adjustment of assembly product volume used for transformation on-device**. To improve transformation efficiency, less of the assembly product is used for transformation. To use 1/5$^{th}$ of the assembly product for transformation, 4 droplets are dispensed away from a sample and the sample is then left to evaporate. To use 1/10$^{th}$ of the assembly product for transformation, first 3 droplets are removed, then the remaining volume is diluted 1:1 with water followed by the removal of 2 droplets and evaporation of the remaining volume.

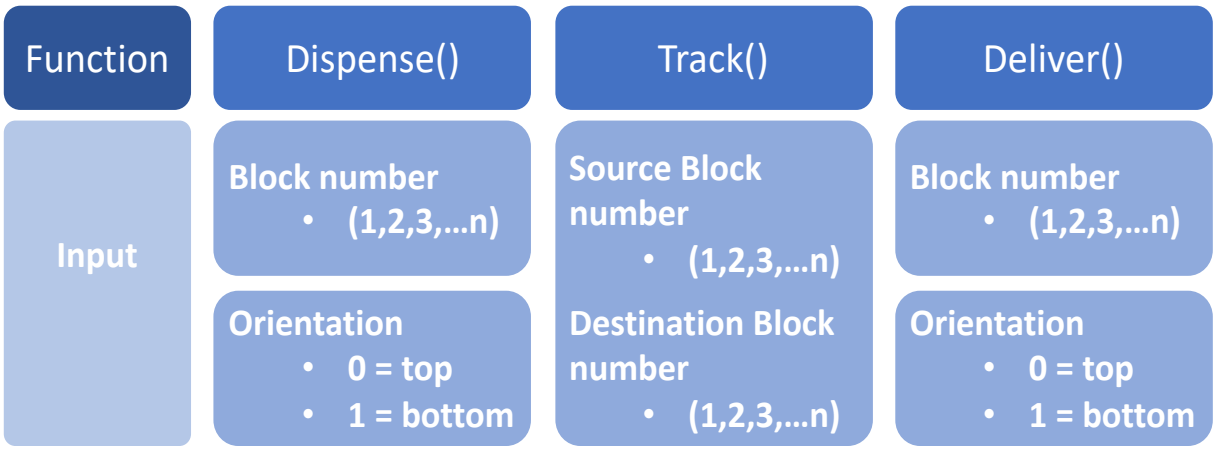



**Figure S7: Modular Python functions designed for modular DMF block units.** The software user interface consists of three main functions: dispense, track, and deliver. For each function, the user can specify which 'block number' to execute the function. For dispense, a droplet is created by moving the stored volume through a neck electrode onto the track and then pulling it between a track electrode and the reservoir electrode. For track, electrodes on the track are sequentially turned on and off to move a droplet across each block of the device. For deliver, a droplet is pulled from a track electrode to a reservoir via a neck electrode. See supplemental videos for further details regarding droplet creation and movement.

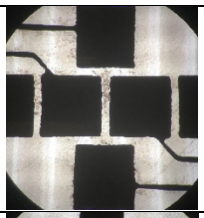
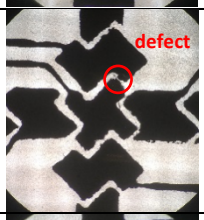
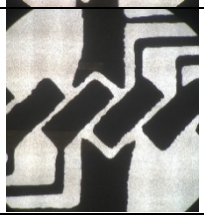
| design | | | voltage used to move droplets ($V_{rms}$) | droplet movement score |
|---|---|---|---|---|
| "square" |  |  | 650 | 0/3 |
| "windmill" |  |  | 200-400 | 3/3 |
| "slash" |  |  | 200-400 | 3/3 |

**Figure S8: Comparison of device designs and their effect on fabrication fidelity and droplet movement.** Three different design approaches were explored. All designs featured 200 µm gap between features. Three devices for each design were fabricated and used to dispense and move droplets across a track of 56 electrodes. Driving voltage for each device test was increased until droplets could dispense and move adequately or until dielectric breakdown occurred. The standard square shape was fabricated easily without defects (connections between features) although the 200 µm gap between electrodes precluded droplets from moving across the entire device in all cases. Increasing the driving voltage past 650 $V_{rms}$ in all cases led to dielectric breakdown. The windmill pattern featured in DMF literature[5,7] was excellent in moving droplets across the device in all cases but several feature connections were identified in all devices. The increased size of the shared border between these electrodes increases the chances of defects (the likely cause due to slight inconsistencies during the toner transfer process). Finally, the "slash" shape has low defect instances similar to the "square" shape, yet this geometry gave excellent droplet movement in all cases. Device defects can be repaired by gently removing the copper using a razor blade.

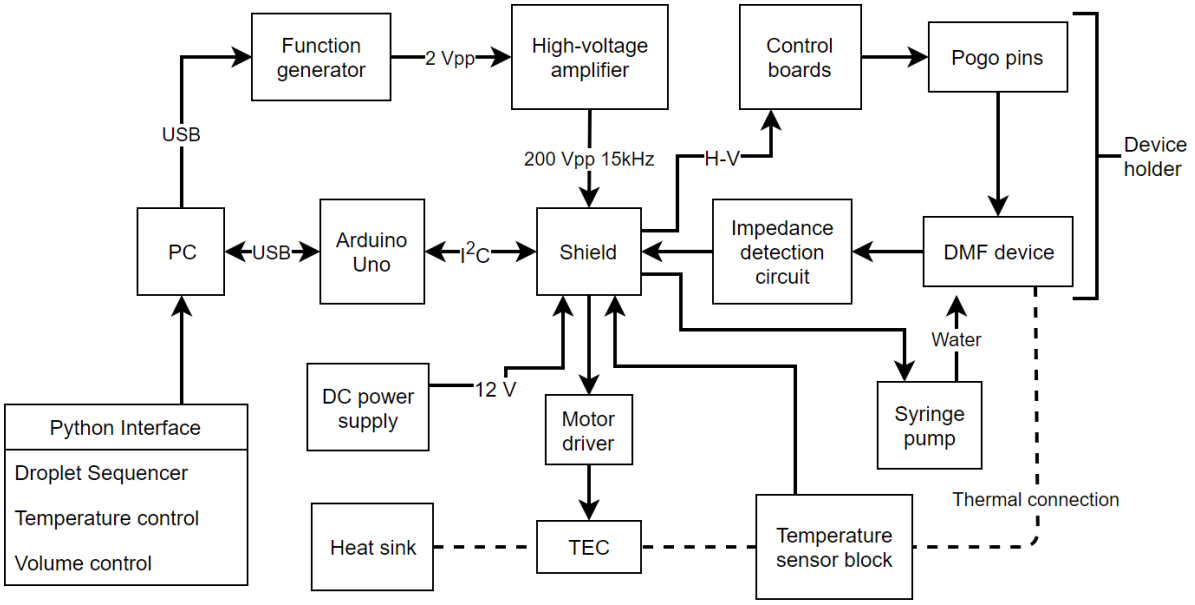



**Figure S9: System setup**. This diagram shows all the connected components involved in the complete DMF system. A custom, in-house shield PCB functions as a central hub for connecting all the electrical components of the system. Thermal connections are accomplished using heatsink compound.

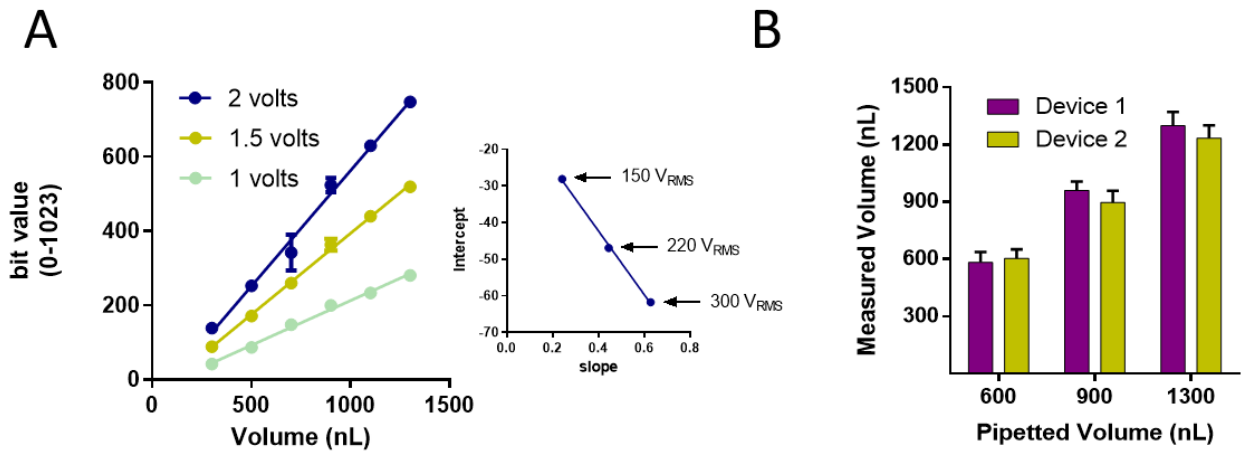



**Figure S10: Volume measurement characterization.** A) Standard curves of volume versus impedance value for three relative voltages were recorded. From this data, a second plot comparing the slope versus intercept for each voltage was plotted (inset). The slope and intercept of this line are used to derive an impedance value-to-volume transfer function given a pipetted volume and its responding impedance value. B) Slight variation between Parafilm M thickness between devices results in different bit values across devices sensed for a given volume. Due to this, devices are calibrated after construction. Two different devices were calibrated with one set of volume and impedance value inputs and their ability to measure 600, 900 and 1300 nL accurately was recorded. An average difference of ~3.1% in recorded volume was determined between devices. We have noticed that much of the variation in this data stems from inconsistencies in pipetted volumes rather than issues in impedance measurement reproducibility (since replicate measurements on device are typically within 2 bits). This method is used to accurately measure volumes on different devices even if a Parafilm-M thickness is slightly different.

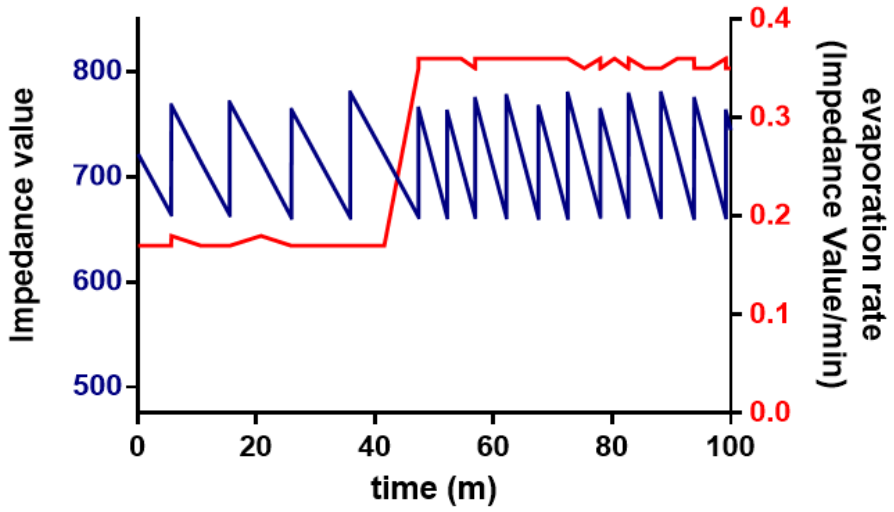



**Figure S11: Python water replenishment system simulation.** The volume replenishment system software component has a simulation option used for development without reliance on DMF hardware. In simulation mode, droplet values and evaporation rates are generated with user-defined vales and degrees of variance. The software was designed to maintain volume values on average at the setpoint by establishing a dispensing trigger threshold equal to ½ the volume of an average droplet. The software calculates the future volume of the sample by using the average measured evaporation rate and decides if dispensing will minimize volume loss given this forecast. Ideally, the system fluctuates by precisely the value of 1 droplet. As depicted, when the evaporation rate changes (as expected with increased temperatures) the system increases the rate of replenishment to maintain sample volumes.

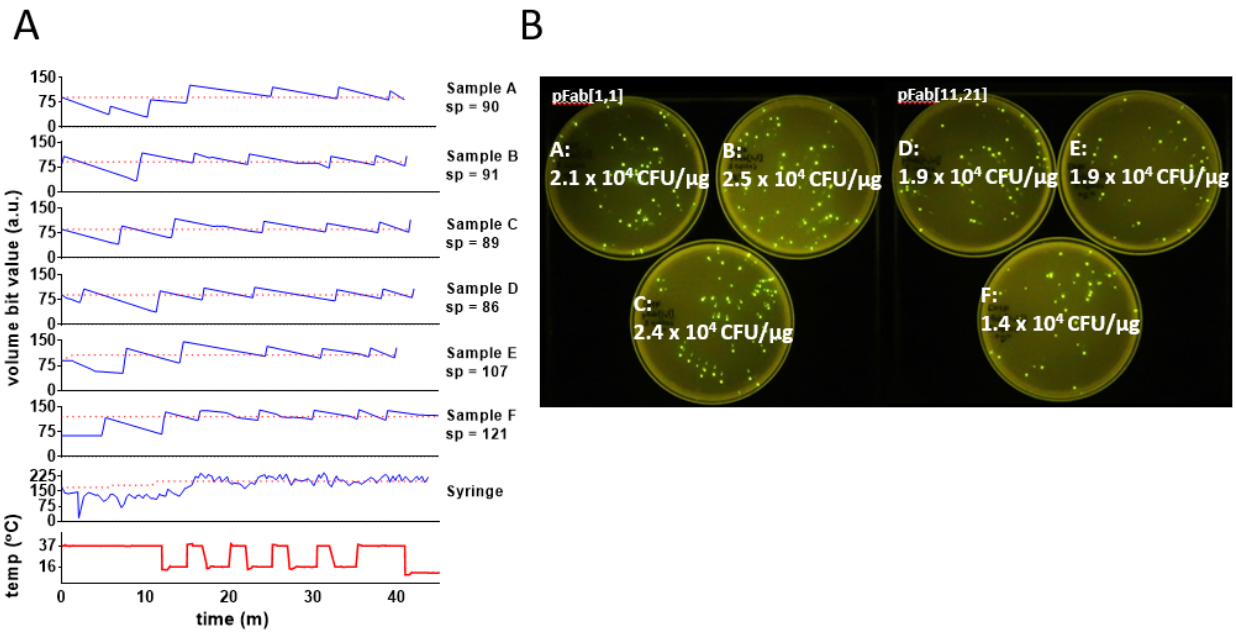



**Figure S12. Five cycles of pFab assembly and transformation on device.** A) Volumes and temperature for the thermocycling component of on-device assembly and transformation. Setpoint (sp) B) Corresponding plate growth for each sample.

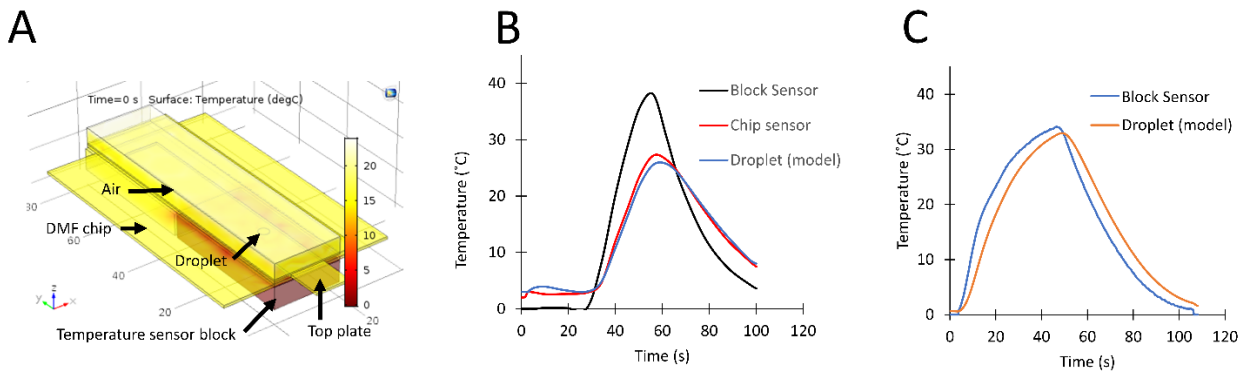



**Figure S13. Device COMSOL model and validation of accurate temperature control and measurement on device.** A) The physical model of the device built in COMSOL. Temperature probes are located at the temperature sensor block and on the device surface. B) Experimental temperature curves for the block sensor and chip sensor compared with a simulated droplet-on-device curve. For this experiment, a larger device-top plate gap height accommodating the chip sensor (3 mm) was considered in the COMSOL model. Here we see nearly perfect agreement between chip sensor and the simulation at this gap height. C) Device block sensor data for a heat shock profile was compared to the predicted temperature profile inside the droplet at the working gap height (150 μm).

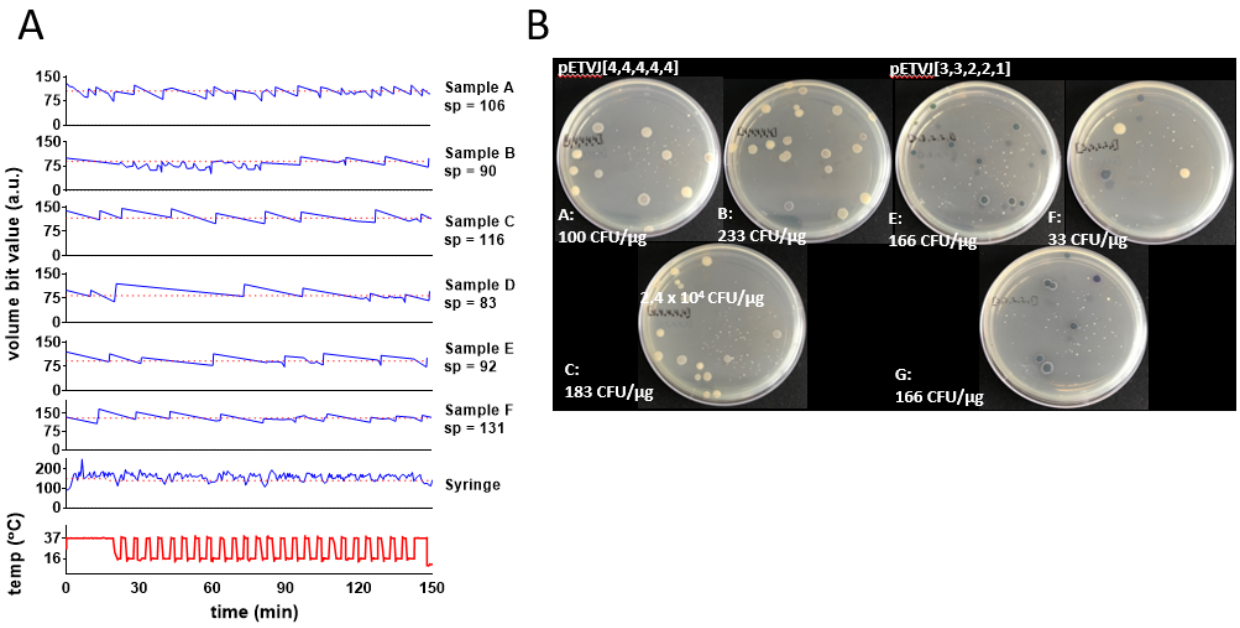



**Figure S14. Twenty-five cycles of pETVJ-VioABECD assembly and transformation on device.** A) Volumes and temperature for thermocycling component of on-device assembly-transformation. Setpoint (sp) B) Corresponding plate growth for each sample. Large, coloured colonies indicate correct assembly. NB: small white colonies are considered background, appearing several days after initial colony formation.

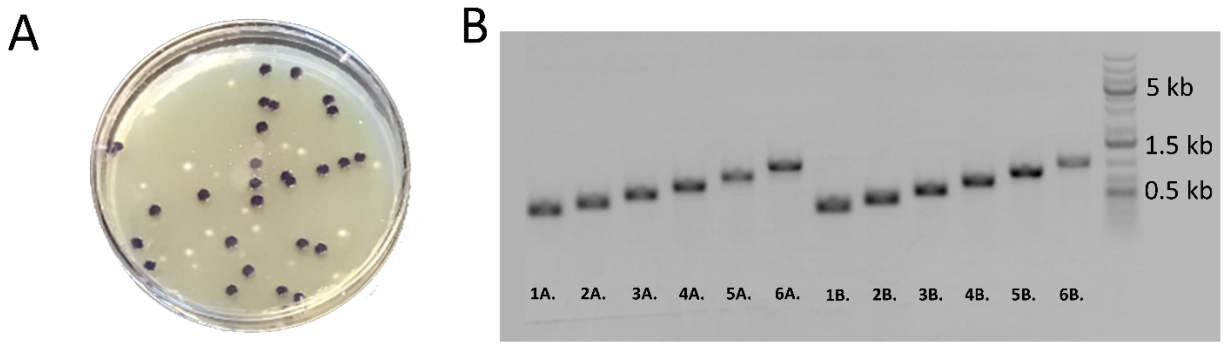



**Figure S15: Verification of pETVJ-VIOABECD assemblies.** A) Plate with BL21(DE3) colonies producing violacein pigment. The presence of purple colonies indicates correct assembly and production of violacein. B) Colony PCR for violacein assemblies. Bands represent junctions between gene cassettes: 1A-6A represent Backbone-VioA, VioA-VioB, VioB-VioE, VioE-VioC, VioC-VioD for all consensus promoters. 1B-6B represents junctions for cassettes with H10-VioA, H10-VioB, H9-VioE, H9-VioC, G6-VioD.

## References


1 J. A. Jones, V. R. Vernacchio, D. M. Lachance, M. Lebovich, L. Fu, A. N. Shirke, V. L. Schultz, B. Cress, R. J. Linhardt and M. A. G. Koffas, *Sci. Rep.*, 2015, **5**, 1–10.

2 R. Green and E. J. Rogers, *Methods*, 2014, 3–6.

3 D. Hanahan, *J. Mol. Biol.*, 1983, **166**, 557–580.

4 E. Moazami, J. M. Perry, G. Soffer, M. C. Husser and S. C. C. Shih, *Anal. Chem.*, 2019, **91**, 5159–5168.

5 S. C. C. Shih, I. Barbulovic-Nad, X. Yang, R. Fobel and A. R. Wheeler, *Biosens. Bioelectron.*, 2013, **42**, 314–320.

6 V. K. Mutalik, J. C. Guimaraes, G. Cambray, C. Lam, M. J. Christoffersen, Q. A. Mai, A. B. Tran, M. Paull, J. D. Keasling, A. P. Arkin and D. Endy, *Nat. Methods*, 2013, **10**, 354–360.

7 I. Barbulovic-Nad, S. H. Au and A. R. Wheeler, *Lab Chip*, 2010, **10**, 1536–1542.

8 S. H. Au, R. Fobel, S. P. Desai, J. Voldman and A. R. Wheeler, *Integr. Biol.*, 2013, **5**, 1014.