Project Title: Plug and play synthetic biology educational resource

Report Title: Progress on the development of a plug and play synthetic biology educational resource

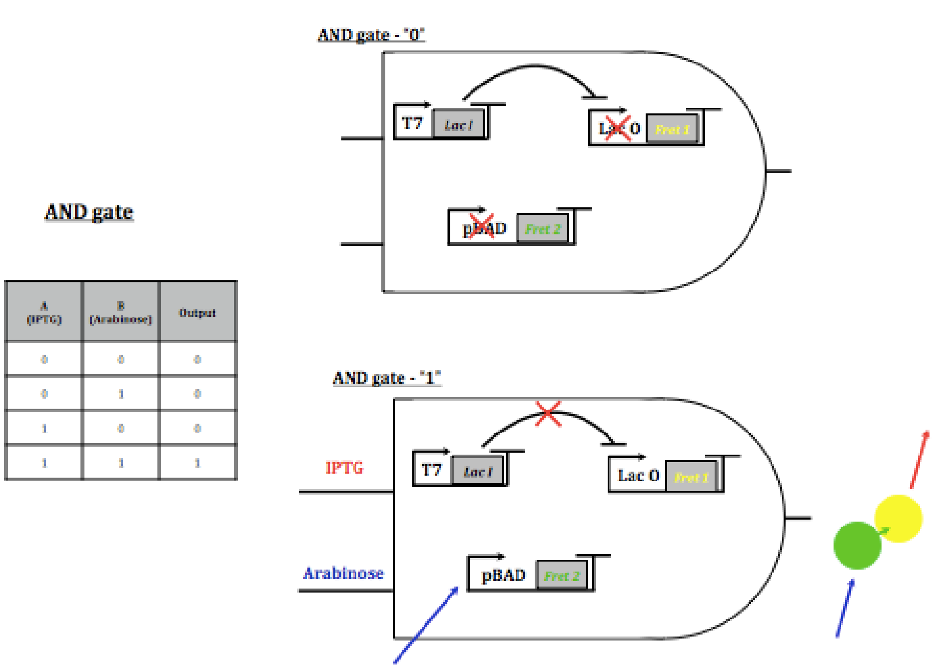
Summary

We set out to develop a prototype teaching resource that will encourage audiences to learn about genetic circuits in a hands on way, providing them with knowledge of the concepts of synthetic biology and the confidence to participate in multidisciplinary synthetic biology research projects at Biomakespace. We brought together an interdisciplinary team of biologists, software and electronic engineers with the aim to develop a plug and play system of physical blocks representing the building components of a genetic circuit. A secondary aim was for the biologists and the engineers to develop an understanding of each other's approach to design requirements of "physical" and "Biological" hardware. To date, we have identified a suitable example of a simple circuit and have obtained a plasmid with the biological components and permission to use this for educational purposes. The team has translated biological requirements for the circuit blocks into software requirements and are currently developing electronic and design requirements of the system. We're developing a workshop where participants build the example physical circuit, which will then be built in biological components in the labs at Biomakespace.

Report and outcomes

With this project, Biomakespace *wants to* develop its first teaching resource to raise awareness, understanding and participation in biology and engineering of biology in the Cambridge area. The timing of this project was ideally suited to coincide with the originally projected opening time of Biomakespace around October 2017 and is one of Biomakespace's first interdisciplinary projects. Due to circumstances beyond our control, the refurbishing of the prototyping space and hence the opening of Biomakespace has been delayed by 5 months – causing a delay in both the lab based as well as physical prototype progress. We have however achieved the following:

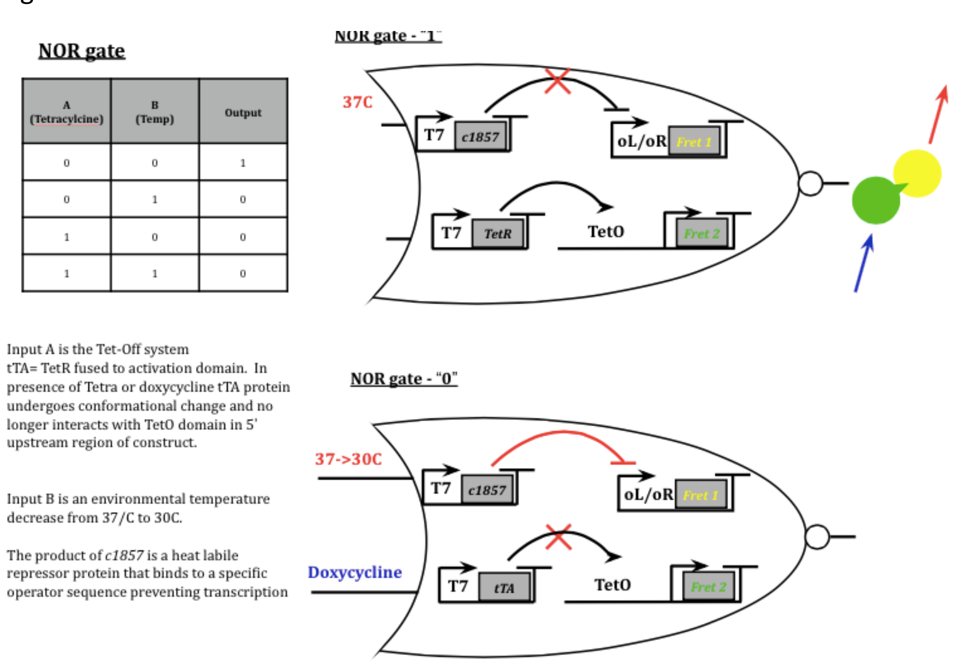
We have identified examples of simple genetic AND and NOR circuits Figure 1a and Figure 1b



AND gate

A (IPTG)	B (Arabinose)	Output
0	0	0
0	1	0
1	0	0
1	1	1

Figure 1a



NOR gate

A (Tetracylcine)	B (Temp)	Output
0	0	1
0	1	0
1	0	0
1	1	0

Input A is the Tet-Off system
tTA= TetR fused to activation domain. In presence of Tetra or doxycycline tTA protein undergoes conformational change and no longer interacts with TetO domain in 5' upstream region of construct.

Input B is an environmental temperature decrease from 37/C to 30C.

The product of *c1857* is a heat labile repressor protein that binds to a specific operator sequence preventing transcription

Figure 1b

Biological workflow – lab based development of the building blocks

In order to start with the simplest of expression cassettes, the team decided to concentrate on the simples inducible system which would later form a component of one of the AND gate circuits. To this end, the arabinose inducible cassette was selected.

Based on the example above, sources of biological components were identified and biological material was obtained from Addgene EGFP-pBAD (this plasmid was a gift from Michael Davidson Addgene plasmid # 54762), an arabinose inducible fluorescence protein (Fig 2 and Fig 3).

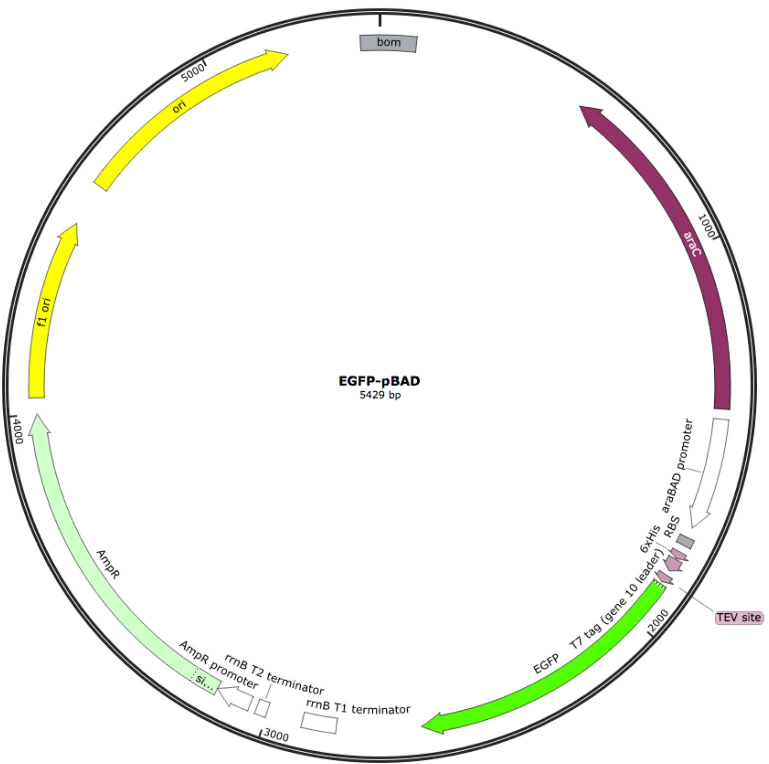


Figure 2. Schematic of EGFP-pBAD vector with EGFP and arabinose inducible promoter labeled.

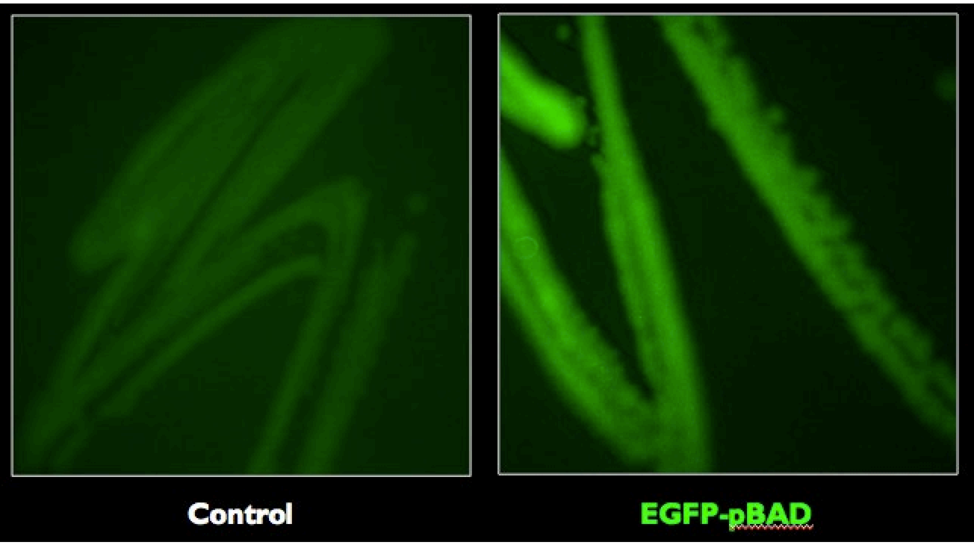


Figure 3. E. coli strains expressing the EGFP-pBAD was confirmed to fluoresce with the expected excitation and emission maxima at 488 nm and 507 nm, respectively.

Future experiments that utlise the EGFP-pBAD construct involve:

1) PCR amplification of the different DNA parts from the plasmid to domesticate for Goldengate assembly and Modular Cloning (MoClo). This allows the team to learn together design, construction and transformation techniques.
2) Expression of EGFP and control of this expression using arabinose. Visualization of EGFP fluorescence will expose the team to different microscopy techniques.
3) This cassette will form one component of the AND GATE circuit (Fig 1).

Around the time of the plug and play project, the synthetic Biology SRI and Science Makers in Cambridge ran a first workshop on DIY circuits at Cambridge's Makespace. Attendees used a cell free system and existing DNA components to build a genetic NAND gate as will as built a DIY 'transilluminator' so the resulting output of the circuit – expression of green fluorescent protein could be visualised. One of the learning points from the workshop was that pipetting small uL volumes with Gilson pipettes is not straightforward to those who have not experienced pipetting before.

This realisation prompted the Biomakspace community to develop a concept for a Molecular Biology 101 course, where all key concepts and techniques to enable modern molecular biology experiments will be covered. From culturing bacteria, DNA isolation/purification, PCR, gel electrophoresis and cloning, inc exercises to pipet accurately and primer design for PCR.

Biomakespace will separately develop the Molecular Biology 101 workshop, which will be a good precursor to building our plug and play circuits in the lab.

Physical representation of the example gene:

The biological requirements for the different genetic building blocks for Arabinose inducible Fluorescence protein gene construct were defined. Once we defined the biological requirements, the software engineers, collaboratively with the biologists defined the software requirements for the blocks. These can be found in Table 1.

	Circuit: Arabinose inducible FRET2			
	Biological part	Biological function	Requirements for the individual parts -biological	Requirements for parts - software
	Small molecule De-repressor: Arabinose (id2.1)	when arabinose binds to the repressor, pBAD promoter gets activated	This is a small molecule. It fits onto a specific repressor protein (specific binding. It can only bind the AraC repressor protein. When it binds, the repressor activity stops, so the promotor changes to ON	**- Circuit2 Derepressor small molecule (id 2.1)** - represents the small molecule (id 2.1) that deactivates the repressor (id 2.2) - physically implemented by a brick that takes sends the output signal (always value = 1)
	Repressor AraC (id2.2)	binds to promoter and stops expression of pBAD promoter	A protein. Binds only to pBAD promoter, not other promoters. When binding to the pBAD promoter-it turns the pBAD promoter off	**- Circuit2 Repressor AraC (id 2.2)** - represents the repressor protein (id 2.2) of the pBAD promoter (id 2.3) - physically implemented by a brick that takes input (D) from the derepressor smmall molecule (id 2.1) and sends its output to the suppressor input (S) of the promoter (id 2.3) - output: !D (negation of derepress input)

Table 1. Biological and software draft requirements for the example gene construct

	pBAD promoter (id2.3)	Arabinose inducible promoter	The part is DNA – so maybe DNA could be represented by a specific colour and promoter (and name) can be written on the part (instead of the different colours I used here for different DNA parts) There is directionality to the part: 5' is to the left or upstream and 3' end in to the right or downstream. The promoter can be constitutively on, either at high or low level, so a dial that regulates amount of signal would be good This part is always upstream from RBS part A repressor (which is a protein) part can bind to the top of the promoter part	**- Circuit2 pBAD promoter (id 2.3)** - represents the pBAD Arabinose inducible promoter of Circuit2 (id 2.3) itself - physically implemented by a brick that takes input (S) from the repressor protein (id 2.2) that turns it off - just by itself, the promoter is ON by default - output: !S (negation of suppress input)
	5'UTR (includes RBS) (id2.4)	Stabilises transcript	The part is DNA There is directionality to this part. 5' upstream and 3' downstream – so directionality indicator required? This part is always directly downstream of promoter and upstream of coding sequence. In terms of functionality in this sequence, it doesn't do anything, it is just there in the right direction	**- Circuit2 5'UTR (id 2.4)** - represents the 5'UTR of Circuit2 (id 2.4) itself - physically implemented by a brick that forwards whatever input it gets from the promoter (id 2.3) to its own output
	FRET2 (id2.5)	Fluorescent protein2	The part is DNA There is directionality to this part 5' upstream and 3'downstream. There are many different genes coding for proteins with different functions (enzymes, chaperones, structural proteins, fluorescent proteins, repressors of gene expression) We need to decide whether and how we will represent these differences	**- Circuit2 FRET2 (id 2.5) coding section** - represents the FRET2 coding section of Circuit2 (id 2.5) itself - physically implemented by a brick that forwards whatever input it gets from the 5'UTR (id 2.4) to its own output
	3'UTR (id2.6)	signals end of translation	The part is DNA. There is directionality to this part. 5' upstream and 3' downstream – so directionality indicator required? This part is always directly downstream of coding sequence (cds) and upstream of terminator part. In terms of functionality in this sequence, it doesn't do anything, it is just there in the right direction.	**- Circuit2 3'UTR (id 2.6)** - represents the 3'UTR of Circuit2 (id 2.6) itself - physically implemented by a brick that forwards whatever input it gets from the coding section (id 2.5) to its own output
	pBAD terminator (id2.7)	signals end of transcription	The part is DNA. There is directionality to this part. 5' upstream and 3' downstream – so directionality indicator required? This part is always directly downstream of 3'UTR. In terms of functionality in this sequence, it doesn't do anything, it is just there in the right direction. pBAD terminator is a diferent DNA sequence than the T7 terminator but function is the same	**- Circuit2 T7 Terminator (id 5)** - represents the terminator of Circuit2 (id 2.7) itself - physically implemented by a brick that forwards whatever input it gets from the 3'UTR (id 2.6) to its own output (this is effectively the final output of the whole Circuit2)

Table 1 continued. Biological and software draft requirements for the example gene construct

Name	Description	Connections	Inputs	Outputs
<derepressor>	- disables the <repressor>	- out - <derepressor> -> <repressor>		- derepress: always 1

<repressor>	- if active (i.e. in absence of <derepressor>), turns off the <promoter>	- in
		  - <derepressor> -> <repressor>
		- out
		  - <repressor> -> <promoter>
			- derepress (signal): D	- supress:
				  - if(D == 0) -> 1
				  - if(D == 1) -> 0

<promoter>	- drives expression of the gene
	- ON by default
	- supressed by presence of active <repressor>
		- in
		  - <repressor> -> <promoter.suppress>
		- out
		  - <promoter> -> <5' UTR>
			~~- enhance (signal): E~~
			~~- enhance weight (dial): We~~
			- suppress (signal): S
			~~- suppress weight (dial): Ws~~
			~~- promoter strength (dial): P~~
				- promoter level:
				  - if(S == 0) -> 1
				  - if(S == 1) -> 0

<5' UTR>	- the region preceding the AUG start codon, which is not translated to protein
	- contains RBS/5'cap, that enables binding of ribosome, so that the mRNA can be translated into protein
		- in
		  - <promoter> -> <5' UTR>
		- out
		  - <5' UTR> -> <coding sequence>
			- promoter level (signal)	- input -> output

<coding sequence>	- the sequence coding for a protein	- in
		  - <5' UTR> -> <coding sequence>
		- out
		  - <coding sequence> -> <state value>
		  - <coding sequence> -> <3' UTR>
			- promoter level (signal)	- input -> output

<3' UTR>	- the region directly downstream of the coding sequence
	- signals end of translation
		- in
		  - <coding sequence> -> <3' UTR>
		- out
		  - <3' UTR> -> <terminator>
			- promoter level (signal)	- input -> output

<terminator>	- signals end of transcription	- in - <3' UTR> -> <terminator>	- promoter level (signal)	none

Table 2. Draft Software requirements for classes of bio bricks.

Currently, the promoter is listed as ON by default to keep it simple. In future, we may have two types of promoter bricks (ON by default, and OF by default)

This step was allotted more time that we originally envisaged as it quickly became clear to all those involved that several discussions were needed to get acquainted with the languages of each discipline and decide how to translate the biological requirements into software requirements.

One specific discussion point was whether the output of a physical gene construct should come out of the coding sequence (cds) brick or whether the output should come out of the terminator block (T).

The electronics and software engineers would opt for the cds brick because this block encodes the functional protein. However, the biologists argued for the output to come out of the terminator sequence as only if all these blocks (including the 3'UTR and Terminator) are correctly assembled will there be an output signal.

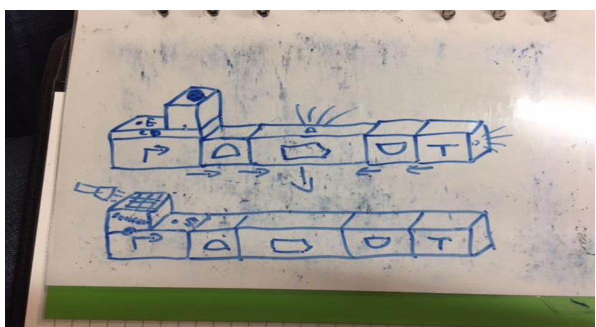


Figure 4**.** Sketch depicting the software engineer's interpretation of output signal coming from the cds block.

The team then progressed to use match boxes to create a physical representation of the bio bricks in order for everyone to understand the biobricks and requirements for each of these. (Figure 5)

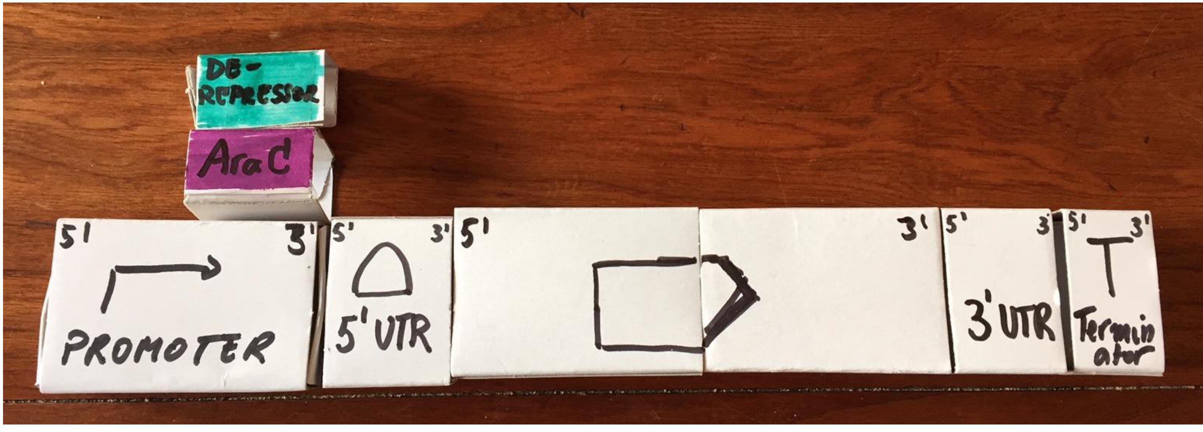


Figure 5. Matchboxes are used to represent the gene construct

Bio brick electronics design considerations

To help make a decision on electronics design, a couple of existing systems were bought to analyse how they work and whether a similar design would be suitable for the purpose of building the plug and play genetic system.

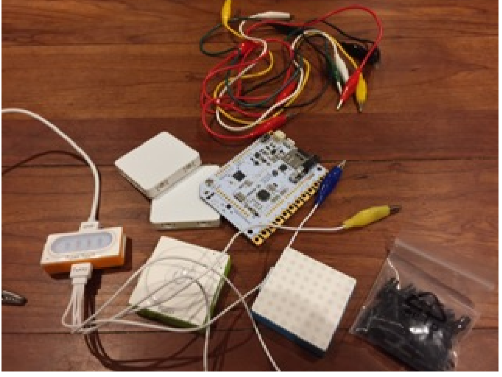


Figure 6. Makeblock Neuron Inventor kit with programmable electronics and a touchboard

The electronics and computing will be distributed in three places:

1. Within each bio brick a relatively low power memory or microcontroller, primarily they report a number or text identifier indicating which biological component they represent. Some intelligence is required so that these values can be programmed for each brick, and to identify themselves at the correct time so that the overall biological circuit can be deduced.

2. A user interface PC, smartphone, or Beaglebone (or similar controller board with sufficient computing power) and LCD display, which device will present text or graphic report on the modelled biological circuit, highlight biology errors and predict the behaviour.

3. A coordinating block that interfaces between [2] and [3]. This will connect to the bricks [1] to enumerate (~a roll call) which bricks are connected to the circuit, in what relative placements, and report over USB, Bluetooth or maybe a custom connection to [2]. This block would distribute electrical power to the bricks [1], perhaps from a battery. Note that this may be integrated with [2] in some of implementations considered.

For the first iteration the plan is to keep the design simple with microcontroller in each bio brick, the user interface software running on a (laptop) PC and communications and power between it and the coordinating block being combined in a USB connection. Ultimately it would be nice to operate with a smartphone, using a Bluetooth radio link, and probably using a rechargeable mobile phone battery (with connection & electronics to work with a compatible charger). [Footnotes a & b.]

For the initial prototype the concept is to put the 'dumb' units [1] inside each brick, have connectors on either end of all/most bricks so that the blocks can be assembled in a chain of bricks. (Missing connections could help guide the user to put together valid assemblies.) The connector will carry 4 or 5 active connections, 2 wires distributing 0V & 5V power, 2 or 3 wires distributing a communication link and controlling so that only one brick communicates at a time. The coordinating block [3] controls the enumeration of the bricks, and produces a map of their connections for the user interface component to assess. A number of connectors can be considered for prototyping to demonstrate the proof of concept, but developing a reusable, robust and reliable connector may need mechanical engineering support.

Other concepts were discussed, and could be tried in a second iteration:

1. Use of RFID tag technology, similar to that used in access cards for building security. All such devices have a 'unique' (within a system) serial number, some of them have a small amount of memory which could be set to a number or text which bio brick they represent. The tags, perhaps in their sticker form, could be placed in each bio brick. The RFID bio bricks could be used with a rollable mat (or rigid board) with a grid of RFID readers, to identify the relative positioning of the bio bricks. This sidesteps this issues with connecting the bricks together, at the tradeoff of complicating the coordinating block to control a number RFID coils. [Footnote c.]

2. Use of a 'breadboard', a rigid board with a grid of connections into which the bio bricks are plugged & unplugged. This could be between A4 and A3 sizes, and there may be a printed guide on the board to help younger students assemble a valid biological circuit, maybe a little like the children's game Operation. One realisation could use USB sockets for the bricks to plug into, with a USB hub under the board powering the USB ports. Each bio brick would have a microcontroller that connects to USB, it wouldn't to do much beyond identify itself as a USB device with the name of the brick's biological function as the device name. The coordinating block would then just use standard USB management function to identify the arrival & removal of the bio brick at each USB port.

Footnotes:
a. Official and third party mobile phone batteries for many brands of phone, e.g. from Amazon. Part selection would be through criteria such as output voltage, simplicity of the charging circuit, wide availability of replacements.
b. Bluetooth tested & approved radio sub-assemblies are available for less than £10, e.g. http://www.hobbytronics.co.uk/egbt-046s-bluetooth-module
c. A rollable RFID sensing mat would be straightforward to develop when we have the Biomakespace prototype lab available for

electronics construction and testing. A similar mat was demonstrated in 2013 by CastAR (an ambitous Augmented Reality system that ran out of development funding in 2017), video here https://www.youtube.com/watch?v=IJrXf6waoUI

Expenditure

Due to delays on opening the Biomakespace, and lack of access to lab and prototype/electronics space, we have not progressed as planned and hence have delayed the purchase of some of the items we budgeted for. We have have currently spent:

Addgene plasmid: EGFP-pBAD	£110.00
Matchboxes	£10.48
Makeblock Neuron Inventor kit	£95.98
Total	**£216.46**

Follow on Plans

Despite the setbacks in opening the Biomakespace and therefore delay in the development of the hardware and lab based resources, this has not tempered the enthusiasm and commitment of the team to develop the resources.

To create the biological building blocks that will be used in future workshops to construct the simple reporter gene sequence example, we will organise a workshop at Biomakespace on the topic of golden gate cloning and primer design to create gene constructs. We will engage our first Biomakespace members to help us design the primers and also amplify the individual genetic sequences for the promotor, 5'UTR, cds, 3'UTR and Terminator. These sequences will then be individually inserted in plasmids and will be our biomakespace building blocks to create our example inducible gene.

Biomakespace is currently open to membership and the first projects (including **Plug and play synthetic biology educational resource)** have started.

Figure 7. The biology lab of Biomakespace

To build the Physical hardware/electronics component of the project, we have already called out to the makespace community to help us with the design of the electronics. With the prototyping lab at Biomakespace expected to be operational by April, we're confident that we will create a larger team that will join in the development of the hardware.

Figure 8. The prototype lab in December 2018 (left) and February 2018 (right)

We will soon be inviting Norwich biomaker community and London Biohackspace to visit Biomakespace and will use that opportunity to discuss the status of the plug and play system and discuss further collaboration on developing and testing of resources.

We will use the remaining funding (£3,783.54) for the activities above and in line with the project budget proposal.

Item	Cost
Events at Biomakespace and travel Bringing together the team that will create the resources, travel between Cambridge, Norwich, London Prototype testing sessions with Biomakespace & London Hackspace members, Norwich outreach network and other interested 3rd parties	£700
Hardware and Materials	£2300
2 Elego sensors kits 2 Little bits kits 10 USB bit-wackers 10 Arduinos 3D printer materials - 7 colours Soldering workstation 1x Bentolab (portable PCR/gelelectrophoresis kit for educational settings)	
Reagents	£1000
~10 plasmids from Addgene (a combination ofregulatory elements, pre-assembled genetic circuits)Examples Include promoters and terminators,repressors and activators,logic gates(incl shipping costs) Sequencing, enzymes, plasmid isolation kits	
Total	£4000

We would like to request £1,000 of follow-on funding to work with product and graphic designers to help create user friendly and attractive resources/blocks and workshop manuals/instructions. £1000 will provide 4 -5 days input of a junior designer.

Changes to team

There are no changes to the team – we do however anticipate that the team will grow in the next few months and we will inform you accordingly