NINT_iGEM plasmid design and construction process – 2008

The goal of this project is to construct connectible logic based on disruption of attenuation structures by anti-sense RNA. Generally three components need to be designed and constructed:

1) Termination/attenuation structures which are inserted in between transcriptional promoters (following the +1 transcriptional start site) and their output to modify transcription. These are, essentially, the logic gates (AND, OR, NOT, NAND, EXOR) that are used to determine whether the circuit produces an output signal or not.

2) Antisense RNA which is inserted following the terminator/attenuator components. These represent the output from one logic circuit and the input to the next circuit.

3) Hammerhead ribozymes flanking the output antisense RNA coding regions. These are small hairpin (stem-loop) structures that act as *cis*-acting (i.e. in-line) enzymes to cleave the output transcript in a specific place. The actual transcript produced from a circuit would normally be composed of the following (in order):

i) Transcriptional start site (+1)
ii) Terminator/attenuator structures plus hybridized RNA input
iii) Hammerhead ribozyme 1
iv) RNA output
v) Hammerhead ribozyme 2
vi) Final transcript terminator

This will be a complex folded structure and could potentially interfere with the subsequent hybridization of the RNA output to the next logic circuit. The Hammerhead ribozymes specifically cleave or cut out the RNA found in between them, releasing just the RNA output for use in the next circuit.

These components (along with others) are incorporated into plasmids so that they can be readily amplified and tested (by growing in the appropriate bacterial cells). The components, themselves, could be constructed in a number of ways. One of the easiest ways would be to have them synthesized completely by one of several companies (IDT, DNA2.0, GeneArt, etc.) that perform this service. Another approach is to simply synthesize one of the DNA strands (or part of it) and use PCR (polymerase chain reaction) to make the complete double-stranded component, using the appropriate primers. In either case the double-stranded DNA encoding the desired component is inserted into the appropriate plasmid using restriction digests and ligation. In addition to the basic components, test harness plasmids need to be constructed so that we can assay the activity of each logic circuit and it various inputs and outputs.

We will follow the BioBrick (http://parts.mit.edu/registry/index.php/Help:Contents) design principles with some modifications to reduce the time required to place the devices into the various test harnesses. BioBrick components contain prefix sequences with EcoRI and XbaI restriction sites and suffix sequences with SpeI and PstI restriction sites. These sites can be used to chain together several components to make a complete functioning component. For example, BioBrick part BBa_R0011 is the pTac promoter, an-IPTG inducible, high-strength promoter. Part BBa_I732018 contains both a ribosome binding site (RBS) and the LacZα coding region (cds). Inserting BBa_I732018 after BBa_R0011 results in inducible high-level expression of the LacZ protein, which can turn the resulting bacteria blue in the presence of the indicator chemical Gal-X.

Both parts are found in carrier plasmids. To insert the second part (BBa_I732018) after the first part (BBa_R0011) in its current plasmid (pSB1A3 – a high-copy, ampicillin-resistant plasmid) requires

a series of restriction digests and a single ligation to glue the components together. We can represent BBa_R0011 as follows:

----- E --- X --- BBa_R0011 (PTac) ---- S --- P ---- term ----

; where E = EcoRI site, X = XbaI site, S = SpeI site, P = PstI site. Note: It is important to realize the XbaI and SpeI are compatible restriction sites. XbaI and SpeI digests can be ligated together, but neither site is recreated and cannot be redigested.

We can represent BBa_I732018 as follows:

---- E --- X --- BBa_ I732018 (RBS::LacZα) --- S --- P ---

Digesting BBa_R0011 with SpeI and PstI opens up the plasmid as follows:

----- E --- X --- BBa_R0011 (PTac) ---- S P ---- term ----

while digesting BBa_ I732018 with XbaI and PstI (and gel-purifying) releases the following fragment:

X --- BBa_ I732018 (RBS::LacZα) --- S --- P

Ligating these together, gives a new plasmid as follows:

----- E --- X --- BBa_R0011 (PTac) --- BBa_ I732018 (RBS::LacZα) --- S --- P ---- term ----

Following construction, we can move the entire component into a low-copy BioBrick plasmid for actual testing (e.g. pSB4A5 – 5 copies, amp^R or pSB4A3 – 10-12 copies, amp^R). We want to make two plasmids that are similar to this one (low copy, amp^R) but with a few convenient modifications. These plasmids will be the workhorses for the project as they will be used to test how various logic circuits and input/output signals are working. The first plasmid is a reporter plasmid very similar to the one described above, but which incorporates the terminator/attenuator logic circuit in front of the LacZ reporter as follows:

----- E --- X --- BBa_R0011 (PTac) --- B --- T/A --- N --- BBa_ I732018 (RBS::LacZα) --- S --- P ----

; where B = BamHI site, N = NcoI site, T/A = terminator/attenuator component (initially null). This should be placed on a low-copy plasmid with ampicillin resistance.

This will allow us to test whether our T/A component is working as designed. We have introduced new restriction sites (BamHI and NcoI) on either side of the terminator/attenuator logic gate. This will simplify our testing procedure as it means we only need to replace the logic gate with a new logic gate (via BamHI+NcoI digest, followed by ligation) in order to make a new test construct. Without this ability, we would have to ligate a new T/A structure after the original BBa_R0011, then ligate BBa_ I732018 after this. This would add several extra days of construction to our plasmid.

If we also introduce a second plasmid, replacing BBa_ I732018 with our output anti-sense RNA, we can begin to connect circuits together.

-- E --- X --- BBa_R0011 (PTac) --- B --- T/A --- N --- HH1 --- H --- oRNA --- Xh --- HH2 --- S --- P ---

; where H = HindIII site, Xh = XhoI site, oRNA = output antisense RNA.

Similarly, in this plasmid, extra restriction sites (HindIII and XhoI) are introduced to make it easy to replace just the oRNA component without having to reconstruct the entire complex plasmid. This component should also be moved into a low-copy ampicillin-resistant plasmid such as pSB4A5.

Constructing Termination/Attenuation components

After the desired Termination/Attenuation component has been designed (using RNAStructure, or its derivative), an IDT ultramer will be ordered representing the DNA-encoding of the T/A component (for T/A components under 200bp). Only one strand (the sense strand should be ordered). The complete double-stranded DNA is made from this by PCR with the common Term+ and Term- primers. These primers are based upon adding BioBrick prefixes and suffixes to the ends of the T/A component, itself. In addition, the NINT-iGEM prefixes also add the BamHI and NcoI sites.

The T/A prefix is: 5' cgatcagaattcgcggccgcttctagacaatagatcattgggatcc
EcoRI EagI XbaI BamHI

The T/A suffix is: 5' ccatggtactagtagcggccgctgcagatgtcc
NcoI SpeI EagI PstI
Extra nucleotides on the distal ends of these sequences assist in digesting distal restriction sites.

The resulting PCR product will be digested with EcoRI+PstI and inserted into an appropriate BioBrick vector (e.g. pSB1A3). This is just for ease of handling and for producing reasonable DNA quantitites of the construct. The construct should be given a valid NINT-iGEM BioBrick name and documented.

In order to test a new logic component, it should be inserted into the BamHI + NcoI sites of the LacZ-containing plasmid described previously. Then it is co-transformed along with one of four signalling plasmids (described below) into an appropriate *E. coli* strain (DH5α or XL1-Blue) and assayed for LacZ activity under all input conditions. As each logic gate is designed and tested *in silico* (using RNA-folding program), appropriate inputs are also designed to activate or inactivate required logic functions. For a logic circuit like 'A AND B' four input test conditions are required: no input, A only, B only, and A and B together. The construction of these test harnesses is described below, following the description of output RNA (oRNA) component construction.

Constructing input RNA (anti-sense) components

In order to ensure that the output from a circuit is a specified RNA sequence, we surround the desired output with so-called hammerhead ribozyme sequences. These are special sequences which cleave the RNA transcript at specific sites. A "null output" component was commercially synthesized which included BioBrick prefix and suffix sequences along with hammerhead ribozyme sequences flanking a random, non-coding signal. This represents the output "0". The oRNA sequence is immediately flanked by HindIII and XhoI sites, incorporated into common sites sharing the same sequence within all oRNA components. This allows us to construct new oRNA components by simply ordering oligonucloetides representing both coding and non-coding ("+" and "-") strands, annealing

them together and inserting them into the HindIII and XhoI sites, replacing the "null output" sequence in a maintenance plasmid (pSB1A3).

After the desired output sequence is designed using the RNAStructure program to test activation of the target logic gate, the following prefix and suffix should be added to the 5' and 3' ends, respectively:

prefix: 5' ccgtcagctcgacaagctttc
HindIII

suffix: 5' aactactcgagatccgtcgacca
XhoI

The resulting sequence can be ordered as two oligonucleotides from IDT (up to 90bp long). They are then annealed in the PCR thermocycler and digested with HindIII and XhoI, then inserted into the same sites in the "null output" BioBrick component in a high-copy plasmid (pSB1A3) for maintenance. Each output sequence is given a unique BioBrick part number and documented. The component is moved into the appropriate reporter vector behind an actual gate for use.

Constructing input RNA (iRNA) testing components

In order to test whether a given logic gate component works as desired, we need to test it under all possible input conditions. For instance a two-input gate 'A AND B' has four input test conditions: no input, A only, B only, and A and B together. In general input RNAs are designed when the gate is designed. The test inputs are assembled into a special low-copy (10-12 copies/cell) plasmid which is arabinose-inducible and has kanamycin resistance (pSB4K5). This means it can be selected and induced seperately from the LacZ reporter which contains the logic circuit or gate being tested. Test iRNAs are designed, ordered and placed into a high-copy plasmid (pSB1A3) for maintenance just like oRNAs. They can then be moved into the test vector (pSB4K5 containing BBa_I0500, an arabinose promoter and repressor cassette) for testing.

The no-input vector simply contains the "null output" oRNA sequence. A or B input vectors will replace this with input A or B iRNA sequences. The combined A and B input vector is simply input vector A with B added at the SpeI + PstI site following.

In order to test, each of the four iRNA constructs (no input, A only, B only, and A and B together) is seperately co-transformed into a test strain along with the test logic circuit. After isolation of transformants, lacZ assays can be performed on six replicates of each iRNA – logic circuit combination in a 96-well plate. Miller units are calculated from the response and documented.

Summary

pSB1A3 (high copy amp^R) will be used for initial cloning and maintenance of indivdiual components. We will make one version for T/A gates and a separate one for i/oRNA sequences.

pSB4A5 (low copy amp^R) will hold test reporters (PLac – T/A – LacZ) and actual circuits with gates and output (PLac – T/A_1 – $oRNA_1$ – T/A_2 – $oRNA_2$).

pSB4K5 (low copy kan^R) will hold test inputs (AraC – PAra – iRNA). Each two-input gate will require four of these: one with "null output", one with input "A", one with input "B" and one with both inputs.