A reconfigurable NAND/NOR genetic logic gate

Angel Goñi-Moreno* and Martyn Amos

Abstract

Background: Engineering genetic Boolean logic circuits is a major research theme of synthetic biology. By altering or introducing connections between genetic components, novel regulatory networks are built in order to mimic the behaviour of electronic devices such as logic gates. While electronics is a highly standardized science, genetic logic is still in its infancy, with few agreed standards. In this paper we focus on the interpretation of logical values in terms of molecular concentrations.

Results: We describe the results of computational investigations of a novel circuit that is able to trigger specific differential responses depending on the input standard used. The circuit can therefore be dynamically reconfigured (without modification) to serve as both a NAND/NOR logic gate. This multi-functional behaviour is achieved by a) varying the meanings of inputs, and b) using branch predictions (as in computer science) to display a constrained output. A thorough computational study is performed, which provides valuable insights for the future laboratory validation. The simulations focus on both single-cell and population behaviours. The latter give particular insights into the spatial behaviour of our engineered cells on a surface with a non-homogeneous distribution of inputs.

Conclusions: We present a dynamically-reconfigurable NAND/NOR genetic logic circuit that can be switched between modes of operation via a simple shift in input signal concentration. The circuit addresses important issues in genetic logic that will have significance for more complex synthetic biology applications.

Keywords: Synthetic biology; Boolean logic; Multifunctionality

Background

The emerging field of synthetic biology [1-7] applies rational engineering principles to the (re)design of biological systems. Work in this area has often focussed on the creation of small-scale genetic devices, such as oscillators [8], toggle switches [9,10], clocks [11], Boolean logic gates [12-15] and half-adders/subtractors [16].

One interesting aspect of such devices concerns their potential for multifunctionality (that is, the possibility that devices may switch between different operating modes, depending on some external signal). Most existing engineered gene circuits have been constructed to perform a single function, but recent results suggest that such devices may be able to implement multiple functions [11,17]. This property is often observed in neuronal networks [18], as it allows organisms to select multiple behavioural "programs" using the same group of neurons.

The ability to engineer multifunctionality into genetic circuits may have significant performance benefits when a range of different responses or behaviours is required. In this paper we describe a model for such a genetic circuit, which may be dynamically reconfigured (without modification) to serve as both a NOR gate (output “1” only when both inputs absent) and a NAND gate (output “0” only when both inputs present), depending on its input. We give the results of single cell computational experiments, before showing how two-dimensional, population-based simulations can shed valuable light on both the behaviour of the system and its beneficial features.

We describe this circuit in the context of our previous work [19] on continuous computation in engineered gene circuits. By “continuous computation”, we mean gene-based computation that maximises the period during which outputs are valid and “readable”, by using “real-valued” signals. This addresses issues of reliability in such circuits, by (a) carefully interpreting binary signal values in terms of continuous/analogue value thresholds over time, and (b) using the concept of branch prediction (taken from computer architecture). During the execution of a
program, a “fork” may occur as the result of a conditional statement (e.g., “if X is true, then do A, else do B”), as in the operation of a logic gate, where the output depends on the inputs. Branch prediction is a technique generally used for saving time when a device faces this kind of decision, and a prediction may be either conditional or unconditional. The latter (studied in [19]) is used when the probability of one branch being taken is significantly higher than the other; in this case, the high probability branch is taken by default, and the situation is only corrected if it transpires that the decision is incorrect, based on the expression evaluation. The circuit proposed here uses conditional prediction, by assuming that the previous output expressed will be carried forward to the next output (and correcting itself if this is not the case) before processing the inputs.

We present our circuit design in Figure 1. Although it can behave as either a NAND or a NOR gate, for clarity we describe here only the NOR logic interpretation of the circuit, and present the multi-functional behaviour in the Results and discussion section. The NOR (negated OR) logic circuit is formed by three sub-components: 1) a logic OR gate, 2) a logic NOT gate (or inverter) and 3) a genetic switch. As the NOR logic function is an inverted OR, the output of the inverter (\(I_2\)) could be taken as the output, along the lines of a classical genetic NOR [20]. However, in our design, the output is denoted by the protein expressed by the switch, (Out), as this allows us to implement branch prediction (that is, changing the switch means that our branch prediction needs to be corrected). In the Results and discussion section we highlight the advantages of the proposed circuit.

The inputs of the circuit, represented by molecules A and B, induce the expression of both genes \(G_1\) and \(G_2\) (by binding to their correspondent upstream promoters), which produce, in turn, proteins \(X\) and \(I_1\) (Inducer_1) respectively. Product \(X\) represses the production of \(I_2\) (Inducer_2), which is expressed by the inverter using gene \(G_3\). Inducers \(I_1\) and \(I_2\) are in charge of controlling the switch, and change its direction. This third part of the circuit (the switch) is formed by two promoters which control the expression of three genes. Repressor \(R_1\) represses the expression of gene \(G_5\) unless inducer \(I_1\) sequesters it, forming the complex \(C_1\) (which has no functionality in the circuit). Symmetrically, \(R_2\) represses the expression of both genes \(G_4\) and \(G_6\) which, in turn, produces the reporter Out.

We now briefly consider the possible implementation of our system. The two main components we use are a genetic toggle switch and NOR gate, both of which have...
previously been successfully demonstrated in the laboratory [9,20]. The main novelty in our proposed scheme (in terms of its implementation) lies in the connection between both components. We believe that this is where attention should be focussed during future laboratory work. The outputs of the NOR gate (that is, the inducers of the switch) are inhibitors of the R1 and R2 repressors. By being sequestered, the repressors are rendered inactive, and the implementation of such a scheme is supported by a recent study [21], in which examples of such negative inhibition are demonstrated. Further investigations may also focus on alternative implementations of the connection scheme, without altering the fundamental behaviour of the device. One possible route to this may lie in directly repressing the switch promoters, instead of implementing the protein-protein interaction. There is also the possibility of using RNA-based logic to implement connections, as described recently in [22]. The key consideration that should inform the engineering process is the fact that the maximum expression level of the NOR output must be at least equal to the maximum expression level of the switch inputs. This is what allows the switch to “flip”. Conversely, the minimum expression level of either NOR output must be lower than the minimum levels of either switch input. These two features allow the device to have the desired multi-functional behaviour.

In terms of traditional electronic logic, our circuit therefore corresponds to a system that produces, by default, a “high”, or “1”, signal in the absence of any input signals equal to “1”. As soon as either inputs equal “1”, the output signal is “pulled low” to “0” (classical NOR behaviour). Once all “1”-valued inputs are removed, the circuit defaults back to “high”. In the Results and discussion section, we study the dynamics of the circuit and - more importantly - the meaning of a logic “1”-valued input, from which the multi-behaviour feature of the system is derived.

Results and discussion

We perform a number of computational simulations (model details are specified in the Methods section), with the two main aims of investigating the behaviour of the branch-predicting NOR gate, and then examining its potential as a reconfigurable device. In both cases, we perform single-cell and population-based experiments, to investigate both the internal dynamics of the circuit and its effect on a spatially-distributed collection of cells (which might be used in a realistic synthetic biology application).

Single cell NOR

We first emphasise the difference between static and dynamic (i.e., continuous) observations. Static measurements are performed by testing a single logic case (input setup), observing from the initial state of the circuit until a steady-state is reached. Dynamic measurements are taken once initialisation has occurred, and the logic input cases are modified sequentially. The outputs obtained are not always consistent, and we conclude that the continuous paradigm is more appropriate (i.e., robust) for these circuits.

Figure 2 shows static observations of the NOR gate. We represent its output value in terms of the concentration of Out: 0 nM corresponds to an input value of “0”, and 5nM corresponds to an input value of “1”. As expected, we only observe an high positive output when both inputs are absent (i.e. zero). In the other cases, although the output is initially expressed due to the constitutively expressed inducer I2, it is soon repressed due to input action.

It is important to note, however, that this performance is only observed when the system starts from a “pristine” (i.e. unused) state. Therefore, this behaviour is only useful if the circuit is intended for “single use”. In non-trivial synthetic biology applications, it may well be the case that
a circuit is used many times, with different inputs, so it is important to test its behaviour over an extended period. Once the system has been initialised with a set of inputs, we therefore need to switch over to a dynamic observation model [19].

In order to study the behaviour of the logic gate over time, we compare the concentrations of inducer \( I_2 \) (which would be a “traditional” NOR output) and our output signal, \( \text{Out} \). Figure 3 shows the concentration of these two proteins over time while the inputs to the circuit are changed dynamically. We observe correct branch prediction, in that an output tends to reflect the previous output. Both \( I_2 \) and \( \text{Out} \) are produced at the outset, when there are no inputs to the system. As soon as one of the inputs is introduced (A, giving an input of 1 after \( t \approx 60 \) hours), the correct output must be “0”. A NOR without prediction, represented here by \( I_2 \), would switch off the expression almost immediately, but \( \text{Out} \) is still expressed for some time (that is, there is a delay in pulling the output signal low, which starts to occur just at the end of the 1-0 input period, and continues through the subsequent 0-1 input period). This is due to the fact that it takes some additional time to “flip” the switch that is controlling \( \text{Out} \), but this delay makes our circuit much more reliable (as we shall see when considering noise). By illustration, unwanted noise in the input will instantly affect \( I_2 \), but the noise needs to be very persistent in order to affect \( \text{Out} \). The same behaviour is observed when the proteins are again expressed (\( t \approx 210 \) in Figure 3), where \( \text{Out} \) delays its expression before returning to “1” (since both inputs return to “0”). In both delays, the system is still predicting the previous output.

**Single cell NOR with noise**

The effect of noisy inputs is shown in Figure 4, where both input concentrations are affected by stochastic noise within different intervals. We highlight in this way the different behaviour of the classic NOR represented by the product \( I_2 \) and our approach represented by \( \text{Out} \). During the 0-0 input phase (until \( t \approx 60 \)) the input values vary within the range \([0 \ldots 0.05]\) nM. This underlines the importance of interpreting binary values in terms of ranges of analogue biological variables. The small changes in \( A \) and \( B \) test the definition of logic “0” in this

| Figure 3 Continuous observations of circuit. Deterministic study of the change of \( \text{Out} \) and \( I_2 \) over time, while the four logic cases are introduced dynamically. Until \( t \approx 60 \) both inputs are “0” (case 0-0); from 60 until \( t \approx 110 \) input A is a logic “1” (case 1-0); until \( t \approx 160 \) input A is “0” while input B is “1” (case 0-1); until \( t \approx 210 \) both inputs are 1; from there onwards both inputs come back to “0”. Logic “0” represented by 0 nM, logic “1” by 5 nM. |
| Figure 4 Continuous observations of circuit, with added noise. Change of \( \text{Out} \) and \( I_2 \) over time while the four logic cases are not homogeneous due to noise in input signals (stochastic inputs). During the case 0-0 (until \( t \approx 60 \)) the logic value “0” varies within the range \([0 \ldots 0.05]\) nM; for the case 1-0 (until \( t \approx 110 \)) input A varies within the interval \([4.5 \ldots 5.5]\) (logic “1”) while input B still varies within the previous interval for a logic “0”; same variation ranges for “0” and “1” during cases 0-1 (until \( t \approx 160 \)) and 1-1 (until \( t \approx 210 \)); From there, again the case 0-0 but with another definition of logic “0”, varying within the range \([0 \ldots 0.005]\). |
In this scenario, the expression of protein still varying within the range \([0, 4.5]\) due to existing repressors in the system. That amount, for our circuit (thus \(Out\) is constantly expressed). That “understanding” is precisely due to the engineered predictive behaviour: as the previous state was 0-0 (i.e., the initial situation), the circuit keeps that state regardless of the noise present.

Dynamic predictions are observed after the 0-0 case, as the initial - static - conditions are no longer valid. During the 1-0 input phase (until \(t \approx 110\)) input A varies within the interval \([4.5 \ldots 5.5]\) nM (logic “1”) while input B is still varying within the range \([0 \ldots 0.05]\) nM (logic “0”). In this scenario, the expression of protein \(I_2\) is completely repressed, as the circuit senses this input as a clear logic “1” -not noise- for both \(I_2\) and \(Out\). The same thing happens during the next (0-1 and 1-1) cases, where the inputs vary within the same intervals for both logic values and output concentrations are the same. In order to get a more valuable insight into the system, we change the meaning for a logic “0” for the final case 0-0 (from \(t \approx 210\)) where it varies within the interval \([0 \ldots 0.005]\). Such a low signal causes the production of \(I_2\) but not at full capacity due to existing \(X\) repressors in the system. That amount, which can be interpreted as a positive output of \(I_2\), it is not enough to change the direction of the switch. Thus, the low inputs (A and B) are interpreted as noise for our system and \(Out\) will stay in the previous state, which is a “0” output. The noise causes an unclear signal to be received at the input: neither a clear logic 0 nor a clear logic 1, thus the system predicts the previous behaviour.

In Figure 5, the results of a full stochastic simulation are shown, using the input profile of Figure 4. The objective is to test the system in a situation where the concentrations of all proteins are subject to randomness. For this purpose, we added Gaussian noise (mean = value, standard deviation = value \cdot noise) at every iteration of the integration. As we see, the overall behaviour remains the same, which allows us to conclude that the logical input values are the key factor determining the correct functioning of the genetic gate. We also observe that the levels of \(Out\) are more distinct (in terms of their mapping onto binary values) than the levels of \(I_2\).

**Population-based NOR**

We now study the behaviour of the circuit inside a population of simulated cells growing on a two-dimensional surface. We use an agent-based simulation approach, which considers the physical factors within the system (cell-cell pressure, collisions, movement, etc.) The first 2-dimensional experiment considers the surface divided in two different areas depending on the inputs they contain (amounts of A and B) as seen in Figure 6. The left-hand side of the surface has both input molecules present (1-1), and the right-hand side has no input molecules present (0-0). As before, the logic “1” concentration is set at 5nM and the logic “0” is set at 0nM. We begin with a single cell in the centre of the surface; cells are “washed out” at the edges, and we assume the constant presence of nutrients (as in a chemostat). In these simulations, we assume \(Out\) to be a green fluorescent protein (for visualisation purposes), and the cell generation time is kept very high (around 12 hours), in order to aid visualisation. That is, because of the delay caused by the switch, we would not be able to observe the desired behaviour at this scale with much lower cell doubling times. Thus, in order to perform the spatial study with a scale that allow us to visualise single cell shapes we increase the doubling time to 12 hours (more details in the Methods section).

We depict the behaviour of the simulated colony in Figure 6, starting with a single cell in the centre of the region. As the number of cells increases, those to the right-hand side eventually exhibit fluorescence, as they inhabit the 0-0 region, while those to the left (in the 1-1 region) show no fluorescence, as expected.

After 50 hours, we notice some cells on the right-hand side that are not producing light, when they should actually display a high output concentration. This is due to the fact that those cells are moved from the left-hand side (1-1 case) at high speed while they are being pushed strongly. Therefore, the circuit inside those cells has not had time enough to respond and start expressing \(Out\) (we recall the gap of Figure 3). After 130h the cells clearly signal the input concentration corresponding to the inputs on the surface (any single-cell “errors” are due to cell movement and/or stochasticity).

We now look at the issue of cell movement in more detail. In Figure 6 we show a population growing in a half
Figure 6 Population-based simulation. Sequential observation of a simulated growing population. The surface on which the cells are growing in contains the inputs with both inputs present in the left half (input logic “1”, established at 1.5 nM for this simulation, as before) and neither input present in the right half (input logic “0”, fixed at 0.0 nM, as before). The output Out is represented as if it were a green fluorescent protein: high expression corresponds to a bright green colour of the cells. The high mobility of cells after 50 and 95 hours (due to there being plenty of free space available) lets us see the graphical pattern produced by the predictive behaviour of the circuit. When the population is very crowded (after 180 hours) the behaviour of the circuit is directly proportional to the surface features. Red circled region: wrong predictions being resolved by changing the direction of the switch. Generation time of cells = 12h.

and half world (in terms of input signal distribution), until we obtain an almost perfect pattern (the outputs matched the inputs, spatially speaking). This precision is obtained due to the low speed (and null direction) of the cells in the centre. In Figure 7 we show the result of a subsequent experiment to investigate the effect on pattern formation of a higher velocity field.

In this experiment, the environment (a longitudinal trap) is divided into two zones: the centre: with a 1/0 input profile, and the remaining area, with a 0/0 input profile.

Figure 7 Effect of cell movement on accuracy. Spatial delay in response due to time spent in changing the direction of the switch. The population is growing from the centre of the longitudinal trap (cells washed out at edges) and the image is taken after 300 hours. The middle sector of the trap (light gray) has only one input (case 1/0) at a high level (4.5nM, which leads to a NOR function), and the remaining area (dark gray) has no input (logic case 0/0). The velocity vector field (lower image) shows the direction and magnitude (colour scheme) of the speeds of every cell at the same time (300 hours).
The cells start growing once inoculated at the centre of the trap (about 100 cells are placed at the beginning). Obviously, as seen in the velocity field (bottom of Figure 7), the cells move in one of two different directions, depending on their physical location: from centre to left, and from centre to right (due to pushing forces while growing). When the cells reach the 0/0 area we would expect a cell's circuit to display a “1” state. This is what we observe, but with a time delay, as seen in the previous differential study. In this particular case, the higher a cell's velocity, the more space it will cover before processing the inputs. This explains the gap between the beginning of the 0/0 area and the region in which the cells start expressing the output Out. This time-space delay plays a very important role in attempts to generate specific patterns in a cell population. If that is the case, there is a key parameter to bear in mind: the velocity of the cells, which can - of course - vary within the same colony (as in Figure 7).

It is important to notice that our circuit offers significant possibilities for pattern formation or sensing studies. Instead of recognising only a logic “1” and a logic “0”, the circuit is also able to distinguish between a high logic “1” and a low logic “1”, and change its behaviour accordingly. We now investigate further this useful property.

**Figure 8** Continuous observations of circuit with lower concentration for logic “1”. Change of Out and I2 over time while the four logic cases are being introduced dynamically. Until \( t \approx 60 \) both inputs are “0” (case 0-0); from there until \( t \approx 110 \) input A is a logic “1” (case 1-0); until \( t \approx 160 \) input A is “0” while input B is “1” (case 0-1); until \( t \approx 210 \) both inputs are “1”; from there onwards both inputs come back to “0”. As before, logic “0” represented by 0 nM, but this time logic “1” is represented by 1.5 nM.

**Figure 9** Effect of different input concentration values. Surface graphs that explore the behaviour of the circuit for different logic “1” and “0” concentrations. For each pair of logic “0” (x axis) and logic “1” (y axis) the experiments shown in Figures 3 and 8 are performed, and the cumulative values of Out and I2 over time are recorded. Those values are depicted in two ways: (1) colour surface (greyscale) with a linear scale from 0 to 700 (low precision as mean values are shown for intervals), and (2) contour lines (colour) with a logarithmic scale for detail behaviour. Output values (surface) shown in arbitrary units corresponding to the cumulative value.
Multi-functional behaviour

In this Section we use different sets of simulations to illustrate one of the main features of the circuit: the possibility of reusing it (without modification) for evaluating a function other than NOR. The key factor lies in how we define input “1”. In contrast to our previous experiments, where this is denoted by an input concentration of 5nM, here we reduce the input “1” concentration to 1.5nM. By “flipping” the high input signal from 5 to 1.5nM, we obtain a change in functionality, from NOR to NAND (negated AND). Such a possibility could prove invaluable in terms of saving space in a hybrid bio-device, if differential behaviour is required for a range of input values.

We compare our approach to that of Budyka [23]; their gates use light as an input, and the functionality of a gate may be altered by changing its wavelength (see also [24], in which the behaviour of a promoter is flipped between that of an amplifier and an OR gate using different inducer concentrations).

In Figure 8 we show the behaviour of the circuit with a concentration of 1.5nM representing input logic “1”, in contrast to the 5nM of Figure 3. We observe how $I_2$ reacts to the changes in exactly the same way as before, thus displaying a NOR behaviour. However, the $Out$ signal now gives the correct output reading for a NAND logic function (which returns 0 if and only if both inputs are 1). When both inputs are introduced ($t \approx 160$), the circuit stops producing $Out$, and does not express it again until

---

**Figure 10** Multi-functional behaviour of the circuit. Spatial cell growth simulation shown, where the inputs are embedded in the surface as follows: top-left quadrant has no inputs, top-right quadrant has input A (1-0), bottom-left has only input B (0-1) and bottom-right has both inputs A and B (1-1). Both rows show the expression of $Out$ over time, but for different logic “1” standards: 4.5 nM for a NOR logic function (top) and 0.5 nM for a NAND logic gate (bottom). Logic “0” is 0.0 nM in all simulations. a.u.: arbitrary units. Generation time of cells = 24h.
the inputs are gone ($t \approx 210$ plus the time needed for the
degradation of $R_2$).

Figure 9 shows the behaviour of $Out$ and $I_2$ over time when
different input concentrations are used as logic val-
ues. We show logic “0” on the $x$-axis of the surface graphs,
and logic “1” on the $y$-axis. The $z$ axis (surface view) repre-
sents the cumulative value of the targeted output protein
($Out$ or $I_2$ depending on the graph) over 300 simulated
hours, while inputs are changed according to the profile
of Figures 3 and 8. For example, if we fix the value of logic
“0” to 0 nM, we observe a change in the concentration of
$X$ when the concentration of logic “1” exceeds 2.8 nM.
When $Out$ abandons the contour line of 316 and enters the
area of 100 (which means it has been expressed for less
time during the 300 hours). However, that change is not
present in the expression of $I_2$, where the scenario is more
homogeneous. This feature is the root cause of the multi-
fuctional behaviour we have just demonstrated.

Multi-functional behaviour in cell populations is shown
in Figure 10, where a bacterial colony grows on a surface
with inputs that are spatially distributed as follows: top-
left quadrant has no inputs (0-0), top-right quadrant has
input A (1-0), bottom-left has input B (0-1), and bottom-
right has both inputs, A and B (1-1). The top row shows
the level of $Out$ when logic “1” is fixed to 0.5 nM, and the
bottom row shows it set to 4.5 nM (in both cases logic “0”
is set to 0 nM). We clearly observe the difference between
the NOR and NAND behaviour of the same circuit placed
in different input scenarios.

Conclusions
The definition of logical values is of vital importance if
different synthetic regulatory networks are intended to
work together. Here we show that a given genetic circuit
can display very different behaviours, depending on the
thresholds of a specific input logic signal. This will be
of significance for future genetic circuit design. The
circuit proposed in this paper harnesses this fuzzy behaviour
by reconfiguring its behaviour between the NAND and
NOR logical functions in response to different input stan-
dards. In this way, the circuit can be reused for either of
those two functionalities without modification. We high-
light the importance of computational studies in order to
find abnormal behaviours inside circuits, and to identify
the key features of a system. Although in this work we
take all parameter values from the literature, the simula-
tion results help us to focus on the specific areas of interest
to parts selection for future laboratory validation.

Methods
Due to the large size of the circuit - with 5 promoters, 10
protein species and 6 genes - the mathematical model is
reduced to 6 Michaelis-Menten equations (4 to 8). How-
ever, a full deterministic model of the first expression
product ($X$) is built in order to: (1) check if the approx-
imation is correct, and (2) make a good setup of the
parameters in equations.

Twelve biochemical reactions describe the expression of
$X$, where both inputs $A$ and $B$ are involved as well as the
gene $G_1$. These reactions are:

\[
A + G_1 \xrightleftharpoons[{k_1}][{k_{-1}}] G_1^a
G_1^a \xrightarrow[{k_5}]{k_5} X + A
B + G_1 \xrightleftharpoons[{k_2}][{k_{-2}}] G_1^b
G_1^b \xrightarrow[{k_6}]{k_6} G_1^b + X
B + G_1^a \xrightleftharpoons[{k_3}][{k_{-3}}] G_1^{ab}
G_1^{ab} \xrightarrow[{k_7}]{k_7} G_1^{ab} + X
A + G_1 \xrightleftharpoons[{k_4}][{k_{-4}}] G_1^{ib}
G_1^{ib} \xrightarrow[{k_8}]{k_8} \phi
\]

where $G_1^a$ denotes the gene with input $A$ bound to its cor-
responding promoter, $G_1^b$ is the gene with input $B$ bound
to the other promoter and $G_1^{ab}$ represents the gene with
both inputs bound. Regarding the rates: $k_1$ and $k_2$ are the
binding rates of A and B, respectively; to their promot-
ers when $G_1$ has no protein bound; $k_{-1}$ and $k_{-2}$ are the
unbinding rates of the previous reactions; $k_3$ and $k_4$ are
the binding rates of B and A to $G_1^a$ and $G_1^b$, respectively;
$k_{-3}$ and $k_{-4}$ denote the unbinding rates of the previous
reactions; $k_5$, $k_6$ and $k_7$ are the active transcription rates
of $X$ by $G_1^a$, $G_1^b$ and $G_1^{ab}$ respectively; $k_8$, $k_9$ and $k_{10}$ are the
degradation rates of $X$, A and B; and $k_{11}$ and $k_{12}$ are the
creation rates of inputs $A$ and $B$ respectively.

With all the rates shown in 1 we extract the following
ordinary differential equations that describe the change
over time of the concentrations of $G_1$, $G_1^a$, $G_1^b$, $G_1^{ab}$, $X$, $A$
and $B$:

\[
dG_1/dt = -k_1AG_1 + K_1G_1^a - k_2BG_1 + k_{-2}G_1^b
\]
\[
dG_1^a/dt = k_1AG_1 - k_{-1}G_1^a - k_3BG_1^a + k_{-3}G_1^b
\]
\[
dG_1^b/dt = k_2BG_1 - k_{-2}G_1^b - k_4AG_1^b + k_{-4}G_1^{ab}
\]
\[
dG_1^{ab}/dt = k_5BG_1^a - k_3BG_1^b + k_4AG_1^{ab} - k_{-4}G_1^{ab}
\]
\[
x/dt = k_8G_1^a + k_6G_1^b + k_7G_1^{ab} - k_8X
\]
\[
A/dt = k_{11} - k_1AG_1 - k_4AG_1^b - k_9A
\]
\[
B/dt = k_{12} - k_2BG_1 - k_3BG_1^a - k_{10}B
\]

All the parameter values used in 2 are taken from stan-
dard values in the literature [25-27]. The objective of this
model is to provide as a generic model as possible, to
abstract away from specific laboratory details (i.e. the util-
isation of a concrete promoter type). Thus, similar kinetic
parameters have the same value in order to prove the fun-
tioning of a complete standardised model. The values are
as follows: $k_1 = k_2 = k_3 = k_4 = 1$ molecules$^{-1}$ hour$^{-1}$; $k_{-1} = k_{-2} = k_{-3} = k_{-4} = 50$ hour$^{-1}$; $k_5 = k_6 = 500$ hour$^{-1}$;
$k_7 = 700$ hour$^{-1}$; $k_8 = k_9 = k_{14} = 0.1$ hour$^{-1}$; $k_{11} = k_{12} = 0.1$
molecules hours$^{-1}$. Notice that $k_7$ is higher than $k_5$ and $k_6$. 

Page 9 of 11
Goñi-Moreno and Amos BMC Systems Biology 2012, 6:126
http://www.biomedcentral.com/1752-0509/6/126
in order to emphasise a stronger transcription rate when the two inputs are bound to their promoters at the same time [28].

With the initial conditions, which are $A = B = 1000$, and $G = 1$, we obtain Figure 11 (left), where we observe the concentration of $X$ over time. In order to simplify the model we approximate the full deterministic model for the expression of $X$ to a single equation. As two promoters control the expression of $X$ (and $I_1$), the promoters can be either additive or can interfere with each other [20]. Considering them to cooperatively interfere, the equation for the rate of change of $X$ over time is:

$$\frac{dX}{dt} = \alpha_X \cdot \frac{[A]^h_1}{K_d_1+[A]^h_1} + \alpha_X \cdot \frac{[B]^h_1}{K_d_1+[B]^h_1} - \delta_X \cdot [X] \quad (3)$$

where $[\cdot]$ denotes concentration. Figure 11 (right) shows the behaviour of this equation over time. As we can see, both Figures 11 (left) and (right) show similar output curves over 30 hours (equivalent initial conditions as explained earlier). All the parameter values of equation 3 have been varied to agree with the behaviour of the full differential model. This validation allows us to express the rest of the system in 5 simplified equations, which are:

$$\frac{dI_1}{dt} = \alpha_{I_1} \cdot \frac{[A]^h_1}{K_d_2+[A]^h_1} + \alpha_{I_1} \cdot \frac{[B]^h_1}{K_d_2+[B]^h_1} - \delta_{I_1} \cdot [I_1] \quad (4)$$

$$\frac{dI_2}{dt} = \alpha_{I_2} \cdot \frac{1}{1 + \left( \frac{[X]}{K_h_2} \right)^h_2} - \delta_{I_2} \cdot [I_2] \quad (5)$$

$$\frac{dR_1}{dt} = \alpha_{R_1} \cdot \frac{1}{1 + \left( \frac{\max([R_2] - [I_1], 0)}{\beta_{R_1}} \right)^h_2} - \delta_{R_1} \cdot [R_1] \quad (6)$$

$$\frac{dR_2}{dt} = \alpha_{R_2} \cdot \frac{1}{1 + \left( \frac{\max([R_1] - [I_2], 0)}{\beta_{R_2}} \right)^h_2} - \delta_{R_2} \cdot [R_2] \quad (7)$$

$$\frac{dOut}{dt} = \alpha_{Out} \cdot \frac{1}{1 + \left( \frac{\max([R_2]-[I_2], 0)}{\beta_{Out}} \right)^h_2} - \delta_{Out} \cdot [Out] \quad (8)$$

where $\alpha$ denotes a synthesis rate, $K$ represents the dissociation constants, $h$ the Hill coefficients, $\delta$ the protein decay or degradation rate and $\beta$ denotes repression coefficients. It is important to notice that the sequenced repressor complexes, $C_1$ and $C_2$, are represented by $\max([R_1] - [I_1], 0)$ and $\max([R_2] - [I_2], 0)$ which are the direct subtraction of the repressor by the inducer (or 0 if a negative result is obtained).

The values of the parameters are chosen to make Figure 11 (right) match Figure 11 (left) according to standard values [26,29]. As before, similar parameters have the same value in order to make the in-silico study as general as possible. Thus, $\alpha_{I_1} = \alpha_X = 3.0$ nM hour$^{-1}$; $\alpha_{I_2} = \alpha_{R_1} = \alpha_{R_2} = \alpha_{Out} = 4.0$ nM hour$^{-1}$; $K_d_2 = K_d_2 = 0.5$ nM; $\beta_{I_2} = \beta_{R_1} = \beta_{R_2} = \beta_{Out} = 0.04$ nM; $\delta_1 = 0.15$ hour$^{-1}$; $h_1 = 1$; $h_2 = 2$.

The initial conditions are: $[A] = [B] = 0.5$ nM (again, both inputs set to a logic 1).

All simulations are performed with our own software coded in Python. Figures 2, 3, 8, 9 and 11 are obtained by using a deterministic approach with the previous ODE (Ordinary Differential Equation) model. For Figure 4 we added noise to inputs, as explained in the Results and discussion section, without changing the ODE model. Figure 5 is obtained by adding noise to the full model (all species that change over time). For that purpose, the ODE model is altered by adding Gaussian noise at every integration step to the previous concentration (building a SDE, Stochastic Differential Equation model). In the spatial studies (Figures 6, 7 and 10) the inputs are fixed in their specific surface areas and the rest of the species are subject to low stochasticity (in this case, however, it is the collective behaviour that matters, not individual).

For spatial studies we use the physics library Pymunk (wrapper for the physics library Chipmunk) to design and control the cells as rigid bodies with growth in an agent-based paradigm. The ODE model of the system is placed inside the cells so there are as many copies of the genetic circuit as cells in a given time (the circuit with all parameter values is copied from mother to daughter when the cells divide). In order to control the doubling time of the cells we can let the circuit run for as many integration steps as we may need before the cell divides. For visualisation purposes, the circuit runs during 12 or 24 hours in a cell life-time (making that time the doubling time) depending on the set-up (see Results and discussion section).

Competing interests

The authors declare that they have no competing interests.

Authors’ contributions

AG-M conceived the study, designed the circuit and performed the experiments. MA supervised the work. Both authors participated in writing the manuscript, and both read and approved the final manuscript.

Acknowledgements

This work was supported by the European Commission FP7 Future and Emerging Technologies Proactive initiative: Bio-chemistry-based Information Technology (CHEM-IT, ICT-2009.8.3), project reference 248919 (BACTOCOM).

Received: 20 July 2012 Accepted: 14 September 2012 Published: 18 September 2012

References

1. Benner SA, Sismour AM: Synthetic biology. Nat Rev Genet 2005, 6(7):533–43. [http://www.ncbi.nlm.nih.gov/pubmed/15995697]

2. Heinemann M, Panke S: Synthetic biology–putting engineering into biology. Bioinformatics 2006, 22(22):2790–9. [http://www.ncbi.nlm.nih.gov/pubmed/16954140]

3. Andrianantoandro E, Basu S, Kaig DK, Weiss R: Synthetic biology: new engineering rules for an emerging discipline. Mol Syst Biol 2006, 2:006.0028. [http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1681505&tool=pmcentrez&rendertype=abstract]

4. Lorenzo RD, Danchin A: Synthetic biology: discovering new worlds and new words. EMBO Reports 2008, 9(9):822–827. [http://www.nature.com/embor/journal/v9/n9/full/embor2008159.html]
5. Moya A, Krasnogor N, Pereto J, Latore: Goethe’s dream. EMBO Reports 2009, 10:528–532. [http://www.nature.com/embor/journal/v10/n1/suppfull/embor2009120.html]

6. Purnick PEM, Weiss R: The second wave of synthetic biology: from modules to systems. Nat Rev Mol Cell Biol 2009, 10(6):410–22. [http://www.ncbi.nlm.nih.gov/pubmed/19616644]

7. Porcar M, Danchin A, de Lorenzo V, Dos Santos VA, Krasnogor N, Rasmussen S, Moya A: The ten grand challenges of synthetic life. Syst and Synth Biol 2011, 5(1–2):1–9. [http://pubs.pmedcentral.nhm.edu/article/index/4512252]

8. Eloizibar MB, Leibler S: A synthetic oscillatory network of transcriptional regulators. Nature 2000, 403(6767):335–8. [http://www.ncbi.nlm.nih.gov/pubmed/10659856]

9. Gardner TS, Cantor CR, Collins JJ: Construction of a genetic toggle switch in Escherichia coli. Nature 2000, 403(6767):339–42. [http://www.ncbi.nlm.nih.gov/pubmed/10659857]

10. Lou C, Liu X, Ni M, Huang Y, Huang Q, Huang L, Jiang L, Lu D, Wang M, Liu C, Chen D, Chen C, Chen X, Yang L, Ma H, Chen J, Ouyang Q: Synthesizing a novel genetic sequential logic circuit: a push-on push-off switch. Mol Syst Biol 2010, 6:350–350. [http://www.pmedcentral.nhm.edu/article/index/5050967]

11. Rodrigo G, Carrera J, Elena SF, Jaramillo A: Robust dynamical pattern formation from a multifunctional minimal genetic circuit. BMC Syst Biol 2010, 4:48. [http://www.pmedcentral.nhm.edu/article/index/5050967]

12. Kramer BP, Fischer C, Fussenegger M: Biologic gates enable logical transcription control in mammalian cells. Biotechnol and Bioeng 2004, 87(4):478–84. [http://www.ncbi.nlm.nih.gov/pubmed/15286985]

13. Anderson JC, Ca Voigt, Arkin AP: Environmental signal integration by a modular AND gate. Mol Syst Biol 2007, 3(133):133. [http://www.ncbi.nlm.nih.gov/pubmed/17648000]

14. Sayut DJ, Niu Y, Sun L: Construction and enhancement of a minimal genetic AND logic gate. Appl and Environ Microbiol 2009, 75(3):637–42. [http://www.pmedcentral.nhm.edu/article/index/5050967]

15. Tabor JJ, Salis HM, Simpson ZB, Chevalier AA, Levsky MA, Marcotte EM, Voigt CA, Ellington AD: A synthetic genetic edge detection program. Cell 2009, 137(7):1272–81. [http://www.pmedcentral.nhm.edu/article/index/5050967]

16. Auslander S, Auslander D, Muller W, Wieland F, Fussenegger M: Programmable single-cell mammalian biocomputers. Nature 2012, 487(7405):272–81. [http://www.ncbi.nlm.nih.gov/pubmed/22690089]

17. Purcell O, di Bernardo M, Grierson CS, Savery NJ: A multi-functional synthetic genetic network: a frequency multiplier, oscillator and switch. PLoS ONE 2011, 6(2):e16140. [http://www.pmedcentral.nhm.edu/article/index/5050967]

18. Popescu IR, Frost WH: Highly dissimilar behaviors mediated by a multifunctional network in the marine mollusk Tritonia diomedea. J of Neurosci 2002, 22(5):1985–93. [http://www.ncbi.nlm.nih.gov/pubmed/11880529]

19. Goñi Moreno, A, Amos M: Continuous computation in engineered gene circuits. Bio Syst 2012, 109:52–6. [http://www.ncbi.nlm.nih.gov/pubmed/22587968]

20. Tamors T, Abar J, Voigt CA: Robust multicellular computing using genetically encoded NOR gates and chemical ‘wires’. Nature 2011, 469(7329):212–5. [http://www.ncbi.nlm.nih.gov/pubmed/21150903]

21. Buchler NE, Cross FR: Protein sequestration generates a flexible ultrasensitive response in a genetic network. Mol Syst Biol 2009, 5(2):272. [http://www.pmedcentral.nhm.edu/article/index/2694680]

22. Wang B, Buck M: Customizing cell signaling using engineered genetic logic circuits. Trends in Microbiology 2012, 20(8):376–384. [http://www.ncbi.nlm.nih.gov/pubmed/22682075]

23. Budyka MV, Potashova NV, Gavrishova TN, Lee VM: Reconfigurable molecular logic gate operating in polymer film. J Mater Chem 2009, 19(41):7721. [http://link.rsc.org/DOI/b008562a]

24. Silva-Rocha R, de Lorenzo V: Broadening signal specificity of prokaryotic promoters by modifying cis-regulatory elements associated with a single transcription factor. Mol BioSyst 2012, 8(7):1950–1957. [http://pubs.rsc.org/en/content/articlelanding/2012/mbs/c2mb25030f]

25. Min-Bueno JM, Rodríguez-Patón A: A simple negative interaction in the positive transcriptional feedback of a single gene is sufficient to produce reliable oscillations. PLoS ONE 2011, 6(11):e27414. [http://dx.doi.org/10.1371/journal.pone.0027414]

26. Gonze D, Halloy J, Goldbeter A: Robustness of circadian rhythms with respect to molecular noise. Proc Nat Acad Sci USA 2002, 99(2):673–8. [http://www.pnas.org/cgi/content/abstract/99/2/673]

27. Dublanche Y, Michalodimitrakis K, Kummerer N, Foglerini M, Serrano L: Noise in transcription negative feedback loops: simulation and experimental analysis. Mol Syst Biol 2006, 2:641. [http://www.ncbi.nlm.nih.gov/pubmed/16200081]

28. Lam FH, Steiger DJ, O’Shea EA: Chromatin decouples promoter threshold from dynamic range. Nature 2008, 453(7192):246–50. [http://dx.doi.org/10.1038/nature06867]

29. Basu S, Gerchman Y, Collins CH, Arnold FH, Weiss R: A synthetic multicellular system for programmed pattern formation. Nature 2005, 434(7037):1130–4. [http://dx.doi.org/10.1038/nature03461]

Cite this article as: Goñi-Moreno and Amos: A reconfigurable NAND/NOR genetic logic gate. BMC Systems Biology 2012, 6:126.

Submit your next manuscript to BioMed Central

• Convenient online submission
• Thorough peer review
• No space constraints or color figure charges
• Immediate publication on acceptance
• Inclusion in PubMed, CAS, Scopus and Google Scholar
• Research which is freely available for redistribution

Submit your manuscript at www.biomedcentral.com/submit