

# Tutorial DACCAD 1.0.x

## 1 Introduction

DACCAD (DNA Artificial Circuits Computer Assisted Design) is a Computer Assisted Design software for Montagne *et al.*'s experimental framework for the building of dissipative reaction networks, dubbed the DNA (Dynamic Networks Assembly) toolbox [1]. Its goal is to help both first comers and experienced users to create quickly DNA toolbox systems, either as a proof of concept or for an actual *in-vitro* implementation. The last section gives some details on how to match the model parameters to your experimental setting.

The DNA toolbox is a set of modules aimed to simplify the design of dynamic DNA computing systems. The basic idea is to keep the system out of equilibrium by continuously generating and degrading short DNA strands. Those strands play a similar role to signaling compounds in biological regulation networks, or to variables in a computer program. They interact with specific longer strands (“templates”) which are kept stable over time and can be seen as the “program” of the system.

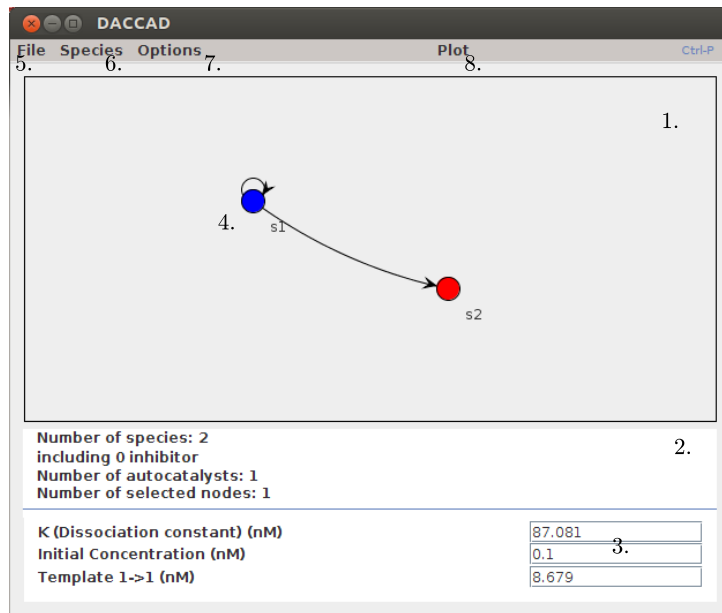
Specifically, the toolbox is composed of three modules: activation, autocatalysis (which can be considered as a special case of activation) and inhibition. Generation is assured by polymerization and nicking, degradation by the action of exonuclease. DACCAD leverages the simple graph representation of DNA toolbox systems to allow the user to easily create and manipulate those systems while automating the ODE generation.

Enzymatic activities and affinities can be set in the options. The default settings are from Padirac *et al.* [2]. See the supplementary material of [1] to experimentaly find values appropriate to your own experimental settings.

## 2 Interface

### 2.1 Main window

This window is used to display the DNA system being designed as well as general information relative to it.



1. Display panel. Shows the graphical representation of the system.
2. Data panel. Shows general data about the system (top) as well as information specific to the selected node(s) (bottom).
3. Parameters area. Allows the user to modify values related to the selected node(s), such as DNA sequence stability or DNA templates concentration.
4. Graph of the system. Nodes represent DNA signal species and arrows DNA templates. Selected nodes are displayed in blue.
5. File menu.
6. Species menu. Groups species related operations.
7. Options. Allows the user to change parameters about the simulator, the enzymes and optimization.
8. Plot. Simulates the system and display the result in a new window.

The display panel represents the system as a graph, following the conventions of section 3. The drawing itself is done by the JUNG library [3]. It features the following possibilities:

- Species (nodes) can be freely added and removed to the system. Activation templates are simply added by first selecting the signal species and then the output species. Inhibitions are added through an option at species creation, allowing the new species to be the inhibitor of an existing module.
- Nodes can be freely moved around by drag and drop. If more than one node is selected, then their relative positions are kept. This change is purely aesthetic and does not change the system itself.
- Additional structures can be added to the current system either by selecting multiple nodes and using the “duplicate” feature, or by importing a previously saved system. In the duplication case, only templates joining selected nodes (as opposed to templates joining a selected node to an unselected one) will be duplicated.

- Selected nodes can be fused together, with a best effort policy to maintain the coherence of the structure. If a module use one of the selected nodes before fusion, it will be replaced by a similar module using the fused species instead. Duplicates are then removed. Fusion is done by a simple flooding algorithm exploring all the currently selected nodes to detect autocatalytic cycles. The species resulting from this operation will have an autocatalytic template if and only if there was at least one autocatalytic cycle. All inhibiting strands targeting a template taking part in such cycle are fused and an inhibitor is created and set to target the autocatalytic template of the new signal strand. All templates inbound toward the selected strands are redirected to the new species. Similarly, all outbound templates are set to initiate from it. This mechanism is useful to quickly combine multiple structures imported from an existing database.
- Right-clicking on a node displays a contextual menu with two options: deleting the node and managing the species inputs, that is the external injections of the species into the system. It provides two parameters: the time at which the injection is made, and the added concentration in case of a spike. Other kinds of inputs can be described in a separate file and imported. Inputs can be periodic, in which case the period can also be set.

The data panel displays the following information:

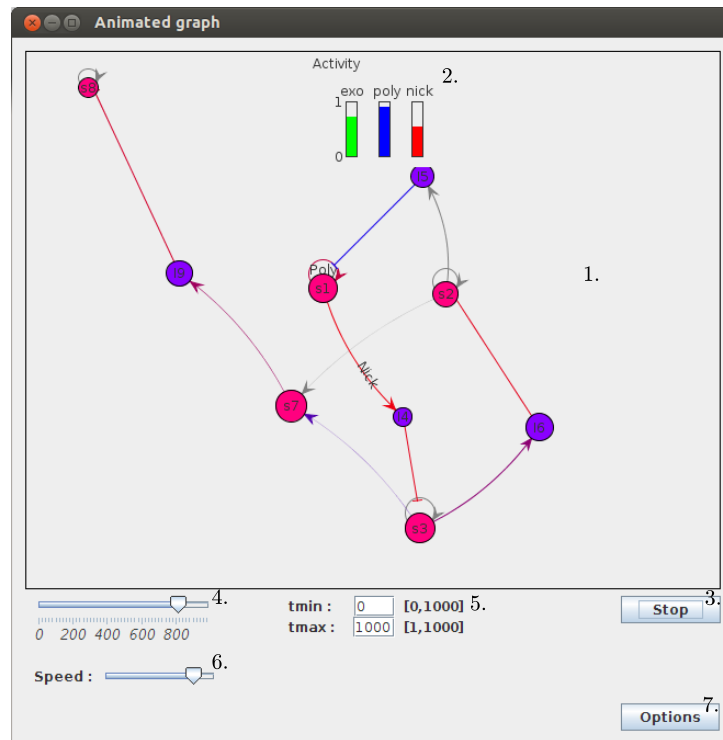
- General data on the system: the total number of species, which can be seen as a degree of complexity of the system, the number of inhibiting species, the number of direct autocatalysts and the number of currently selected nodes.
- Contextual data, depending on the selected nodes. If one species is selected, its initial concentration and  $K_s$  (dissociation constant) are displayed and can be adjusted. If this species has an autocatalytic module, the concentration of template of this module can also be set. If two species are selected the concentration of template of modules using one as signal and the other at output are displayed, with a maximum of two if both species are activating each other. The edition area also contains an button opening a window with additional options, such as deleting the template or setting its coaxial slowdown (default 0.2).

The option menu allows to set all relevant parameters to simulate the system, notably the activity and Michaelis parameters of all enzymes along with the possibility to declare them saturable and to toggle competitive coupling, the  $K_{duplex}$ ,  $\alpha$  and simulation time. It is also possible to declare which species concentration should be displayed in the time plot and set which parameters can be modified by the optimiser.

Finally the plot button commands the simulation of the system using the Gragg-Bulirsch-Stoer integrator from the Apache Commons Mathematics Library. The resulting time trace is then displayed using the JFreeChart library [4] in a separate window. It can also be exported as a text file to be used with other programs, such as Matlab or Gnuplot.

The remaining actions are standard (“new”, “save”, “open”) or straightforward (“export”).

## 2.2 The animate window



1. Display of the graph. Nodes radius: concentration of the species. Nodes color: stability as expressed by the dissociation constant, blue is stable (dissociation constant low), red unstable (dissociation constant high). Edges thickness: total template concentration. Edges color: quantity of loaded or active template (that is, anything that is not free or inhibited), gray means less than 10% active, then the scale goes from blue to red. Nick, poly, exo: each mark represents which template is saturating said enzyme the most. If a given enzyme is saturated to the point of changing the behavior of the system, those marks can be used to find the bottleneck.
2. Saturation panel: shows the current observed activity of each enzyme. 1 means the enzyme is available at full speed (no substrate), 0 means the enzyme is completely saturated.
3. Animate/Stop button: starts (respectively stops) the animation.
4. Time: represents which moment of the simulation is displayed.
5. Minimum and maximum time: sets which part of the simulation should be animated.
6. Speed scale: sets the animation speed.
7. Options: allows to set some appearance options, such as whether to display the enzyme saturation marks or how to calculate the nodes radius.

## 3 DACCAD tutorials

This section gives four tutorials presenting the possibilities of DACCAD.

Note: if at any time you make a mistake or would like to go back, you can use the undo option from the "File" menu (shortcut Ctrl+Z). It is also possible to remove nodes with right click (secondary click on Mac OS)>remove. Another possibility is to select the nodes and press "Del". Templates can be removed by selecting the nodes it connects and pressing the "X" button in the data panel.

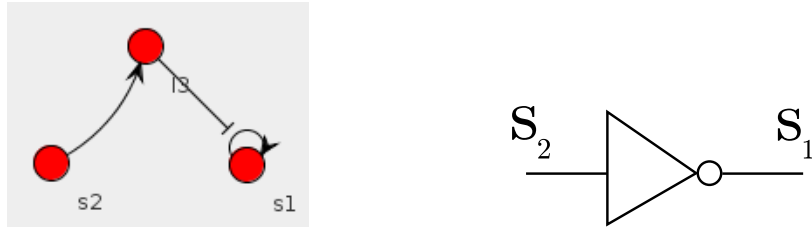


Figure 1: Left: inverter gate made with the DNA toolbox. Right: logical diagram.  $s_2$  is the input and  $s_1$  the output.

Similarly, enzymatic parameters can be reset, using the “default” button in the “Enzymes” tab of the options. Default parameters for sequences dissociation constants are (roughly) 74 nM for a signal species and 0.2 nM for an inhibiting species. Templates have a default concentration of 10 nM, with 0.2 coaxial slowdown.

### Tutorial 1. A simple logical gate: the NOT operation

This tutorial will highlight:

- the creation of signal species and templates.
- parameter setting.
- pulse inputs.

Logical circuit make great use of the NOT gate. Its working is simple: when it has an input, no output should be produced. Conversely, in the absence of input, there should be an output. In the DNA toolbox, an inverter (or NOT gate) is made of an autocatalyst that can be inhibited by the action of a second signal species. The diagram of the gate and its logical equivalent are shown in Figure 1.

**First step:** create a new activation species. This can be done through the “New species” function in the “Species” menu (shortcut Ctrl+A). In the pop-up window, choose “activation species”. A new species named  $s_1$  should appear. Select it by clicking on it (it will become blue) and change its initial concentration (last line of the data panel) to 1 nM.

**Second step:** Add the autocatalysis template. Adding a template creating a species  $b$  from a species  $a$  is simple: select  $a$  and then click on  $b$ . In the present case,  $s_1$  should already be selected, so it is enough to click on it again. The graph should look like the one on Figure 2, left. You can simulate this network by pressing “plot” (shortcup Ctrl+p), which should yield a nice steady state after transient amplification. If the curve is completely flat, check that the initial concentration of  $s_1$  is not zero.

**Third step:** add a new activation species (Figure 2, center). It is possible to reposition it by drag-and-drop.

**Fourth step:** add a new species, select inhibition. An inhibition species always targets a specific template, so the application will open a pop-up window to ask which template should be inhibited. Note that, due to restrictions in the DNA toolbox, only templates generating signal species can be inhibited<sup>1</sup>. In the present case, there should be only one choice (“1->1”), press “ok”. Add a template from  $s_2$  to the inhibitor (Figure 2, right). The NOT gate is now complete.

To test the behavior of the NOT gate, we can add a pulse input on the species  $s_2$ . This input represents an injection of  $s_2$  in the system.

<sup>1</sup>To work around this limitation, it is possible to use an indirect activation. That is, a signal species activates another signal species which in turn creates the inhibitor. It is then possible to inhibit the first activation.

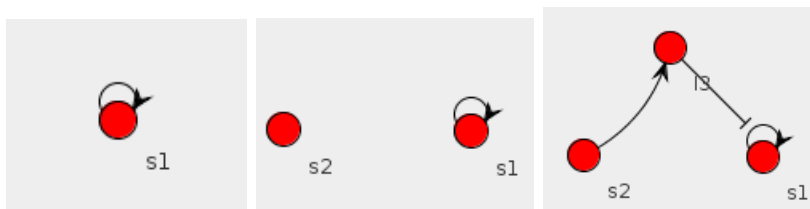


Figure 2: NOT gate: step by step creation of the network. The output is shown in dark blue, inputs respectively in red and cyan.

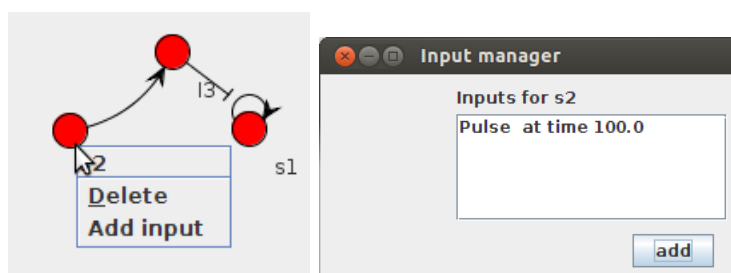


Figure 3: Adding pulse inputs

To open the input manager, right click (secondary click on Mac OS) on the species  $s_2$  and select “add input” (Figure 3, left). This should open a new window. Just click add and close (Figure 3, right). It is also possible to modify input parameters (such as input time, quantity or periodicity) or to delete an input by selecting an existing input in the input manager. Species can have any number of inputs, which are all managed through the input manager window. Species with inputs are displayed double-circled.

We can then use the plot option (shortcut  $\text{Ctrl}+\text{p}$ ) to observe the simulated behavior of the system. We obtain indeed a signal inversion (Figure 4). Note that it is possible to automatically resimulate and replot a system every time a change occurs by toggling “autoplot” on in the “Options” menu.

Finally, we can save this system in convenient place, using the “File>Save” command.

## Tutorial 2. Creating a more complex gate: the AND gate

This tutorial assumes that the first tutorial was completed. It focuses on:

- multiple selection and multiple nodes reorganization.

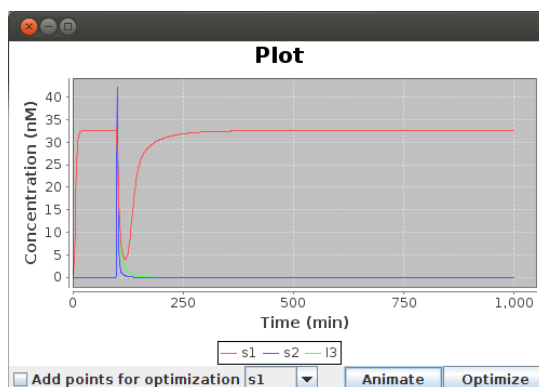


Figure 4: Behavior of the NOT gate. When  $s_2$  is injected at time 100, the output signal  $s_1$  is drastically reduced, before returning to normal when  $s_2$  is hydrolysed.

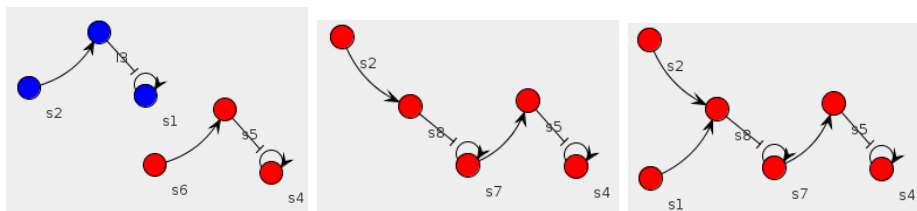


Figure 5: Creation of the logical circuit.

- advanced graph manipulation: the “join” and “duplicate” operations.
- stability modifications.
- continuous inputs.

In this tutorial, we will realize an AND gate, building on the previous NOT gate. We will then add continuous chemical inputs to the logical inputs of the circuit to observe its response to external signal. In a wet lab, such continuous input can be realized by using microfluidic devices [5].

**First step:** starting from the NOT gate, either by opening a previously saved file or after the previous tutorial, select all nodes. To do so, you can click on an empty space and drag the pointer across the screen to perform a multiselection, or use Shift + left click to toggle which nodes are selected. Once all nodes are selected, use the “Species>Duplicate selected”. Drag and drop the selected nodes so that the network looks like Figure 5, left.

**Second step:** Click on an empty spot to clear the selection. Select the output of the first gate and the input of the second gate (respectively  $s_1$  and  $s_6$  on Figure 5, left). Use the “Species>Join selected” operation to fusion those two nodes. Reorganize the network as in Figure 5, center. Set the initial concentration of all autocatalysts to 1 nM.

**Third step:** Simulate the system with the plot option. If there is no pulse at time 100, add a pulse input on the circuit input ( $s_2$  here). We can see that the pulse is not strong enough to propagate to the output of the circuit ( $s_4$ ). Indeed, the species gets degraded before it was able to have enough impact. To solve this problem, it is possible to use more stable species. Select  $s_2$  and change its dissociation constant to 4 nM. The lower the dissociation constant, the longer it takes the species to detach from a template. After plotting again, we can see that there is now a large impact on the output.

**Fourth step:** We can also use this inertia to create an AND gate. This gate should have an output if and only if both its input are present. To do so, we will create a new activation species (see tutorial 1) and connect it to the first inhibitor (Figure 5, right). Set the dissociation constants of both species to 20 nM. Replot. You can see that there is a noticeable impact on the system, but no output is created. Add a pulse input to the new species at time 100. Replot. There is now a spike of output in the system after a delay. The whole set of possibilities is shown in Figure 6 and is the same as a logical AND gate.

**Fifth step:** finally, we can add an external continuous input representing an flow of chemical species, like in an microfluidic open reactor, for instance. To do so, create a simple file in a convenient place. Edit it with a program that will not add formatting, such as Emacs or Windows Notepad. Each line of this file should contain a value representing the added quantity in nM at the  $n^{th}$  minute, where  $n$  is the line number. In our case, copy and paste zeros for approximately 40 lines. This will leave enough time to the circuit to reach its stable state. You can then add any value for the next couple of lines. Floating point and negative values are accepted. If the file is longer than the simulation time, extraneous values are ignored. If the file is too short, it is padded with zeros. Then return to DACCAD and choose “Species>Import input”. Add the input to  $s_2$  and replot. Species can have any number of inputs, which are all managed through the input manager window.

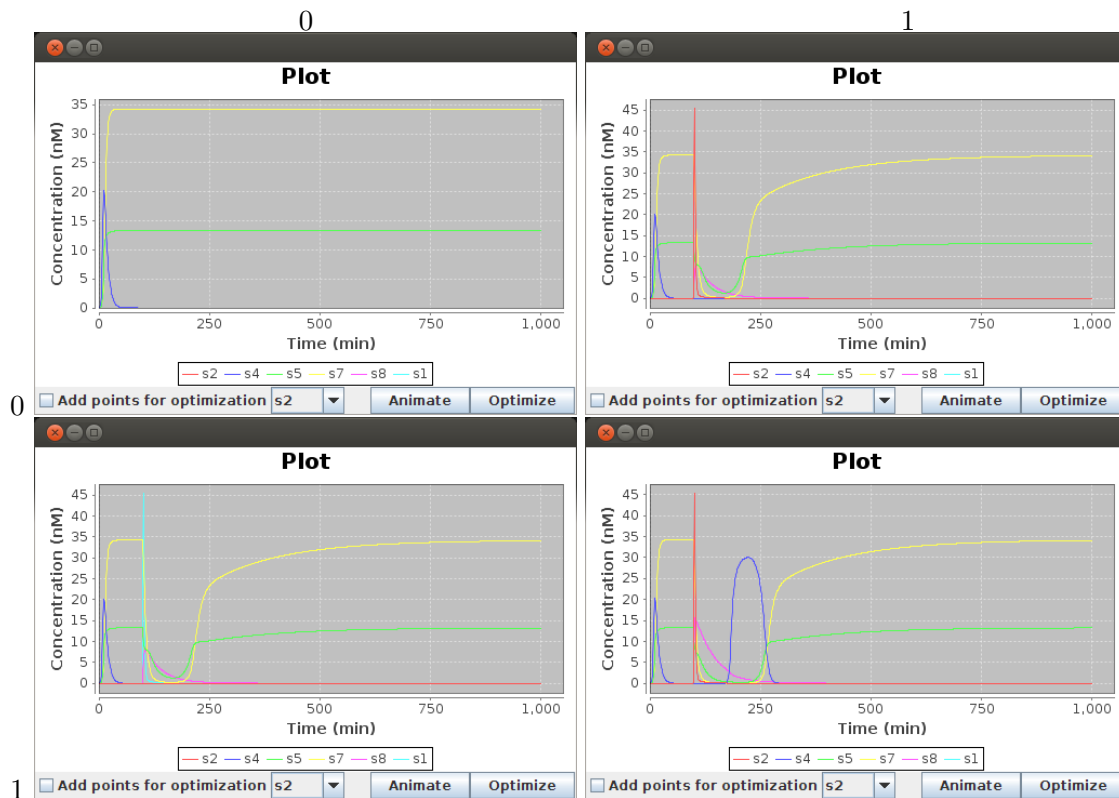


Figure 6: Truth table of the AND gate.

### Tutorial 3. Importing modules and animation

This tutorial requires that you completed the first tutorial and saved the resulting system. This tutorial focuses on:

- setting templates concentrations.
- importing previously saved modules.
- the animation view of DACCAD.

In this tutorial, we will use an oscillating scheme to modulate a downstream module and give it a spiking behavior. Indeed, it is often necessary to create spikes, to switch bistable for instance. We will then observe and understand the resulting behavior with the visualization tool of DACCAD.

**First step:** This tutorial will create a large system, which is subject to enzymatic saturation. While it would be possible to make the current system robust against saturation, it is beyond the scope of the present tutorial. Moreover, when designing a system, it may be better to first turn off enzymatic saturations, check the integrity of the design, then toggle the saturations back on and optimize the parameters. To turn off the saturations, go to “Options>Enzymes” and turn all “saturable” checkboxes off. Now enzymatic activity is assumed to be first order.

**Second step:** Follow the pattern show in Figure 7, left. Start all the autocatalysts by setting the initial concentration of  $s_1$  to 5 nM,  $s_2$  and  $s_3$  to 1 nM. Plot. You should obtain the results shown in Figure 7, right.

**Third step:** Use “Species>Import module” to import the previous NOT gate. The import operation allows you to add a previously saved system without overriding any parameter and is useful to quickly assemble a large system made of basic modules. Move the added gate to a convenient spot on the screen.



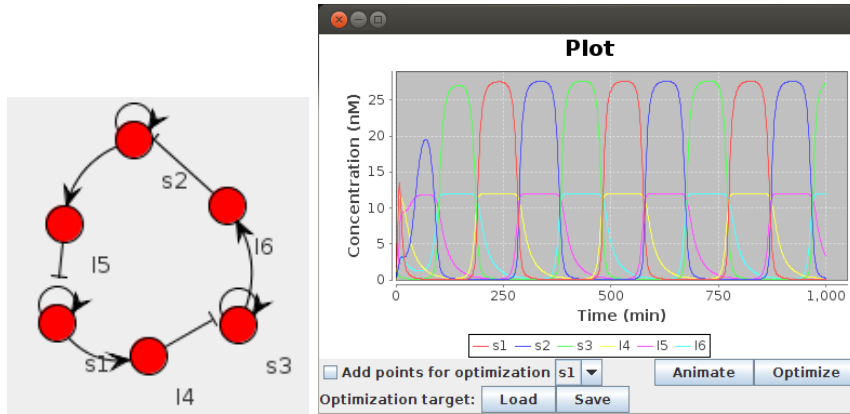


Figure 7: Padirac's switch oscillator [6].

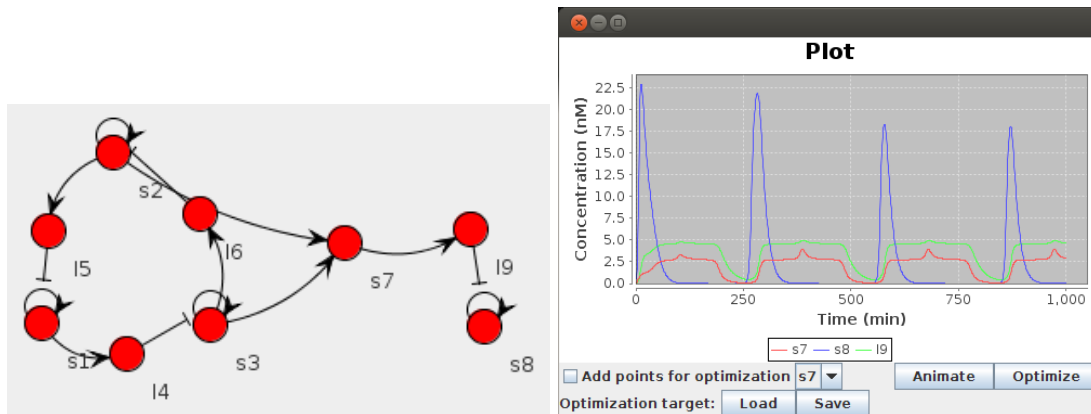


Figure 8: The spike generator.

**Fourth step:** Connect both  $s_2$  and  $s_3$  to the input of the gate (Figure 8, left).

**Fifth step:** The current system generates too much inhibitor, which destroys the output. Select both  $s_2$  and the input of the gate ( $s_7$  here). You can do so by using Shift + left click to add a species to the current selection. Then, set the template concentration to 1 nM. Do the same with  $s_3$  and  $s_7$ . Also set the template from  $s_7$  to the inhibitor of the NOT gate to 5 nM. You should observe the result shown in Figure 8, right. You can also try different combinations of the template concentrations to change the shape and intensity of the spikes.

**Sixth step:** Toggle the saturations back on. You will notice that the shape of the plot is not optimal anymore, with large spikes. One way to change this is to play with the coaxial stacking slowdown of the templates. This effect represents the capacity of two DNA strands to stabilize each other when they are only separated by a nick. While not as stable than a full strand of the same length, the strands still denature slower than when alone on the template. This slowdown is sequence (and thus template) dependent, and can be accessed through the options of a given template (button on the right of the template concentration in the data panel). Based on multiple sources [7, 8], realistic values seem to be between 1.0 (no slowdown) and 0.04 (25 times slowdown).

Once you are satisfied with the plot, you can observe the dynamical evolution of the system by pressing “Animate”. In the new window, press “Animate” to start the animation. You can get a better visualization for this system by opening the options, setting the radius scale to “linear” and the radius size factor to 0.3, for instance.

## Tutorial 4. Optimization

This tutorial is independent of the previous ones and explains how to use the optimization tool. It is recommended to have done the first tutorial to know the basics about graph creation and edition.

We will realize a simple system with an autocatalyst activating an output, then use the optimization tool to change their respective levels.

**First step:** Create the system. Add an autocatalyst, set its initial concentration to 1 nM, add an activation species and finally add a template from the autocatalyst to this new species. The resulting graph is shown in Figure 9, top.

**Second step:** When plotting, you can see that both steady-states are identical and that both curves are similar. We can use CMA-ES [9] to set a different level for both species. However, we don't want to evolve all possible parameters, we only want to change the template concentrations. Go to "Options>Optimization", select "Species dissociation constants" and uncheck the "optimize" option. Do the same with "Initial species concentrations". Since we are evolving two different species at the same time, it is a good idea to get a stricter (that is lower) threshold ratio. 0.5 should be a good starting point. The threshold ratio defines how close to the user-defined points the plot has to be on average. Sigma represents the initial standard deviation used by CMA-ES to modify the evolved parameters. If the fitness (that is the distance to the target) does not change much with respect to a given parameter, the algorithm will increase this parameter's sigma. Conversely, if the fitness changes fast, the algorithm will reduce the parameter's sigma.

**Third step:** Close the options. If you closed the plot window, plot again. Our goal is to get 40 nM of output for 20 nM of input. To change the display range, right-click on the plot, then go to "Properties>Plot>Range Axis>Other>Range". Uncheck the "Auto-adjust range" box and select a maximum of 45. Then close the properties window, check the "Add points for optimization" box and add two point for  $s_1$  at the 20 nM level. Then use the drop box to select  $s_2$  and add two points at the 40 nM level. The resulting screen should look like Figure 9, bottom left. If you added a point by mistake, you can always remove it by clicking on it. You can also add points by loading a previously saved optimization file. The loaded points will be added to the target of the species currently selected.

**Fourth step:** Press "Optimize". After a short while, the species steady state should be correct (Figure 9, bottom right). You can cancel the optimization at any time by closing the optimization window. The optimization will then stop at the end of the current generation. You can then select either  $s_1$  alone or both  $s_1$  and  $s_2$  to check the updated templates concentrations.

## 4 File formats

This section describes the multiple file formats. While not explicitly human readable, they can be edited by hand, or generated by another resource.

### 4.1 Graph files

Graph files should have the ".graph" extension and contains three mandatory sections and two optional ones. It is possible to include commentaries, as the parser will ignore any line starting with a sharp symbol (#). The sections are as follow, and should appear in this order:

- SEQ: describes the species that are present in the graph. A line in this section contains the following data, separated by a tabulation: species ID, species initial concentration, node x coordinate in the layout, node y coordinate, sequence stability.

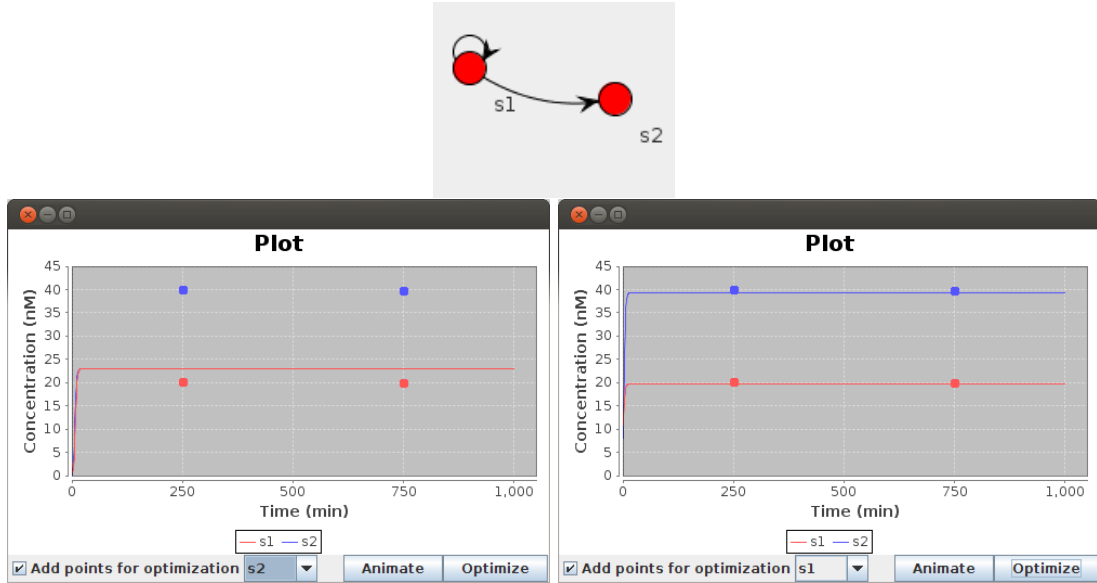


Figure 9: Optimization of parameters.

- TEM: the templates of the graph. The values represent the name (purely aesthetic, the name displayed in the data panel), input species ID, output species ID and total concentration.
- INHIB: which species are inhibitors, and of which template. Inhibiting species ID, template name (should be the same as in the previous section), template input species, template output species.
- INPUTS: optional, describes species inputs. Species ID, type (“pulse” or “file”), type specific data (time and concentration for a single pulse as well as frequency for a periodic pulse, URL for a file).
- PARAMS: optional, gives the system parameters. Parameter name, value. Each line is optional, and can be given in any order. The parameters abbreviations are:
  - absprec: absolute precision for integration.
  - relprec: relative precision for integration.
  - inhfact: stability ratio between an inhibitor on its template (fully double-stranded) and on the template it inhibits (double-stranded with mismatches). The higher this value, the less stable the inhibitor when inhibiting. This value cannot physically be inferior to 1, but this is not prevented by DACCAD.
  - diplrat: polymerase activity slowdown when displacing the output of a template. Only used for long outputs (i. e. inhibitors).
  - exokmib: exonuclease Michaelis-Menten parameter ( $K_m$ ) with respect to inhibiting species.
  - exokmsi: exonuclease Michaelis-Menten parameter ( $K_m$ ) with respect to signal species.
  - exokmtm: exonuclease Michaelis-Menten parameter ( $K_m$ ) with respect to free templates. Even though they are not degraded, they competitive inhibitor of the exonuclease.
  - exovm: exonuclease reaction speed. Enzymatic activity is defined as  $\frac{V_m}{K_m}$  with the relevant  $K_m$ .

- kdplex: hybridization speed of complementary single strands.
- nickkm: nicking enzyme Michaelis-Menten parameter ( $K_m$ ).
- nickvm: nicking enzyme reaction speed. Enzymatic activity is defined as  $\frac{V_m}{K_m}$ .
- maxtime: simulation time in minutes.
- polkm: polymerase Michaelis-Menten parameter ( $K_m$ ) with respect to templates without output.
- polkmb: polymerase Michaelis-Menten parameter ( $K_m$ ) with respect to templates with both input and output.
- polvm: polymerase reaction speed. Enzymatic activity is defined as  $\frac{V_m}{K_m}$  with the relevant  $K_m$ .
- selfsta: polymerase self-start ratio. Represents the activity slowdown when the polymerase extends a template without primer.
- toehold: hybridization slowdown when a template input or output displaces an inhibitor.

Here is a minimal working example:

```
SEQ
1 0.0 44.446860742124116 95.64789361292605 74.32767762106766
TEM
1->1 1 1 10.0
INHIB
```

## 4.2 Input files

Input files are simple text files with a value per line. The  $n^{th}$  line represents the flux (in  $\text{nM}\cdot\text{min}^{-1}$ ) at the  $n^{th}$  minute, values for the flux in between are computed using quadratic interpolation. Note that negative values are acceptable. If the file is not long enough for a given simulation (less lines than simulation minutes), it will be padded with zeros.

## 4.3 Optimization files

Those files are used to store optimization profiles. Each line represents an optimization point, composed of two values separated by a tabulation character. The first value represent the targeted time, the second the targeted concentration. Note that this file does not specify which species is the target, leaving it as a choice for the user. If multiple species should be optimized at the same, the various profiles have to be stored in different files.

## 4.4 Exported files

There are three types of exported files:

- raw simulation data: each line represents the concentrations of all species in the system at a given time (one minute per line), separated by a tabulation character.
- SBML file: exports the whole system as an SBML file, which can be used with other applications, such as Copasi.
- Mathematica file: exports a slightly simplified version of the system as a Mathematica file. Some options are not available yet in this format (inhibitor displacement and polymerase self-start), but will be added soon.

## References

- [1] K. Montagne, R. Plasson, Y. Sakai, T. Fujii, and Y. Rondelez. Programming an in vitro dna oscillator using a molecular networking strategy. *Molecular systems biology*, 7(1), 2011.
- [2] A. Padirac, T. Fujii, and Rondelez Y. Bottom-up construction of in vitro switchable memories. *Proceedings of the National Academy of Sciences*, 109(47):E3212–E3220, 2012.
- [3] Joshua O'Madadhain, Danyel Fisher, Padhraic Smyth, Scott White, and Yan-Biao Boey. Analysis and visualization of network data using jung. *Journal of Statistical Software*, 10(2):1–35, 2005.
- [4] David Gilbert. The jfreechart class library. *Developer Guide. Object Refinery*, 2002.
- [5] Jean-Christophe Galas, Anne-Marie Haghiri-Gosnet, and André Estévez-Torres. A nanoliter-scale open chemical reactor. *Lab on a Chip*, 13(3):415–423, 2013.
- [6] Adrien Padirac. *Tailoring spatiotemporal dynamics with DNA circuits*. PhD thesis, Université Claude Bernard Lyon I, 2012.
- [7] Vadim A Vasiliskov, Dmitry V Prokopenko, and Andrei D Mirzabekov. Parallel multiplex thermodynamic analysis of coaxial base stacking in dna duplexes by oligodeoxyribonucleotide microchips. *Nucleic acids research*, 29(11):2303–2313, 2001.
- [8] Dmitrii V Pyshnyi and Eugenia M Ivanova. The influence of nearest neighbours on the efficiency of coaxial stacking at contiguous stacking hybridization of oligodeoxyribonucleotides. *Nucleosides, Nucleotides and Nucleic Acids*, 23(6-7):1057–1064, 2004.
- [9] Nikolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 312–317. IEEE, 1996.