Using fuzzy logic in synthetic biosystems design

Yves GENDRAULT1,2, Morgan MADEC2, Martin LEMAIRE2, Christophe LALLEMENT2

1 ECAM Strasbourg-Europe - 2 rue de Madrid - 67300 Schiltigheim

2 Laboratoire Icube – UMR 7357 – Equipe SMH

300 bd Sébastien Brant - BP 10413 - F-67412 Illkirch Cedex

Jacques HAIECH3

3 Laboratoire d’Innovation Thérapeutique (LIT) - UMR 7200

74 route du Rhin - F-67400 Illkirch

Contact author: ygendrault@unistra.fr

*Abstract* — *In silico* Design steps of synthetic biosystems are only slightly discussed in the literature but they could benefit from the expertise of other areas where they are mastered for many years, such as microelectronics. The adaptation of the design flow of digital system for synthetic biology is being carried out by our team and, in this context, we propose a computational core of fuzzy logic adapted to this field to provide quantitative results in automated design step. In a previous paper we demonstrated the value and adequacy of fuzzy logic in modeling biological systems. Here, we present optimizations and performances of our algorithm on a simple example of a biological AND gate, then we illustrate its application to the automated design of an XOR gate.

# Introduction

The field of synthetic biology has emerged a little more than a decade ago and aims to combine biotechnology with engineering sciences to create novel synthetic biosystems [ref]. Meanwhile, microelectronics is an area where the computer-aided design of systems has proven itself for many years, thanks especially to an effective methodology. Porting this methodology to synthetic biology will bring a lot to this science.

The approach used by our team is to reuse existing tools from microelectronics and to adapt them to the biological material. We thus start to recreate an entire environment dedicated to synthetic biology to assist in the development of new biosystem designs, based on similarities between the standardized elements of synthetic biology, the BioBricks [ref], and basic digital electronics components [ref].

Starting with a behavioral description for the target, the tools developed allow us to automatically generate an assembly of BioBricks, closest to this description. This step is based on a logical modeling of biological mechanisms and on algorithms integrated with standard logic synthesis tools used in microelectronics. The biosystem thus obtained is then modeled using ordinary differential equations and also simulated with specific microelectronics or biology tools to verify the accuracy of its behavior.

The link between these two levels of modeling is not easy to achieve in biology, because there are multiple biological mechanisms that respond to the same logic function. To fill this gap, we had the idea to use fuzzy logic, which should allow a finer selection of possible BioBricks assemblies in the design step, by providing quantitative results.

We had previously presented the development of models of intermediate level of abstraction in fuzzy logic [ref]. The value and accuracy of fuzzy logic for biological modeling has been shown and seems promising for the design step. The purpose of this paper is to go further and to automate the design of biosystems thanks to fuzzy logic modeling.

At first we remind the basics of fuzzy logic, and we present the improvements made to our algorithm. We then validate its use on a simple system and analyze its performance in terms of various parameters. Finally, we present the automation biosystem design with fuzzy logic based on two examples.

# Fuzzy logic

Classical logic involves only two possible states: the Boolean state "true" and the Boolean state "false." Fuzzy logic, introduced by Zadeh in 1965 [ref] is closer to the field of probability by considering that a state has a degree of membership to previously defined intervals from the total range of values.

The algorithm for performing fuzzy logic operations can be divided into three main parts. The first step, called fuzzyfication, can convert the input data (often quantities) of a set of suitable data to fuzzy logic employed by checking the membership of these to the reference intervals, called membership functions (MFs). These functions can be of several types, but generally triangle functions are used. Then, the data constituting the output of the system are computed from the input data according to a set of rules provided by the user. This second step is called rules evaluation. The rules can again be provided in several forms, but we use matrices, which correspond to the model of the mechanism itself. Finally, the output of the system is expressed in a quantitative manner and is calculated from the output data in the last step, the defuzzification. This general framework was applied on biological modeling [ref] and the evolutions developed are detailed in the next section.

# Optimisation of Fuzzy logic algorythm

Our previous algorithm was coded on Matlab [ref], which is well known for matrix computation and very efficient regarding simulation times. Models developed in fuzzy logic are coded in the form of matrices of rules so the choice of Matlab appeared appropriate. However, the software need some libraries to work and the loading time of them just lengthen the total simulation time. In addition, the language remains closed-source and its integration into automated design flow is likely to be complicated.

We have thus developed our algorithm in C language, which allows full compatibility with other tools.

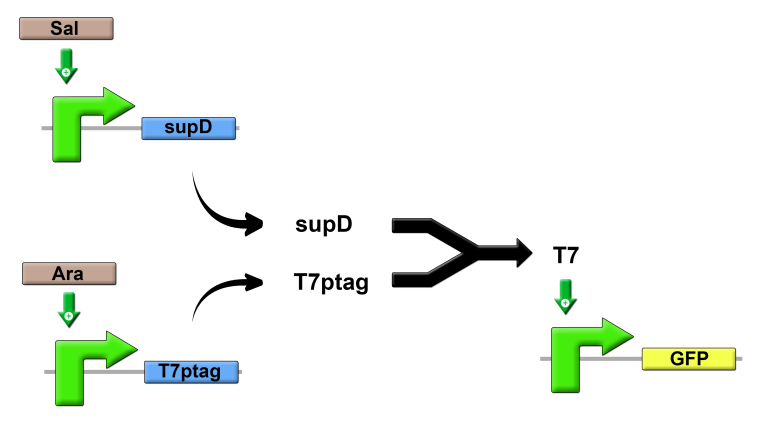
Paragraphe Martin sur la réalisation technique du soft en C.

With this new code, the execution time of the algorithm is reduced from 10-5 to 5\*10-8 seconds for calculation of a single point with a precision of five MFs, while being compatible with the functional synthesis step of our design flow [ref]. We also conducted several tests on the performance of this algorithm on synthetic biological systems, which is developed in the next section.

# validation and performances analysis

To test the robustness and efficiency of our algorithm, we conduct several tests: the link between the number of MFs and the accuracy achieved and the impact on the computation time, the cascading of several models and the effect of standardization on results. The mechanism used for these tests should correspond to a relatively simple component. We selected the AND gate from Anderson *et al.* [ref].

## AND gate and testbench



1. AND gate from Anderson *et al*.[ref].

This gate has two inputs, *Sal* and *Ara*, who allow respectively the synthesis of *supD* and *T7ptag*. These two proteins can bind to form *T7*, which is the activator of the GFP reporter protein. GFP protein is synthesized only when both inputs are present which is indeed the behavior of an AND gate.

To represent this gate using fuzzy logic, we start with the experimental results provided in the publication and we created the following matrix of rules:

(Matrice à fournir par Martin).

## Measuring performance

Plusieurs tests ont été réalisés pour étudier l’algorithme :

Effet du nombre de MFs sur le temps de simulation et la précision des résultats

Script pour remplir les matrices automatiquement en fonction du nb de MFs et d’une matrice de départ (Brève description de la méthode par Martin)

Etude t simu vs finesse de modélisation en fonction du nb de MF : 3 -> 100

Résultats Martin : Courbe des différents temps de simulation en fonction du nombre de MFs + justesse du fit

Nombre intéréssant en terme de ratio temps/précision semble être 7 MFs

Effet de la normalisation

Méthode de calcul de la logique floue impose une variation de la sortie non comprise entre 0 et 1 mais légèrement au-dessus de 0 et en-dessous de 1. Nécessité de normalisation pour des systèmes rebouclés.

Résultats courbe plus étiré lors d’un nombre de MFs faible.

A développer

Effet de la mise en cascade

Explication de la méthode de test et résultats de Martin

Nb de MF nécessaires doit être au moins égal ou supérieur au nombre d’étages.

## Search algorithm of the matrix of rules

A développer par Martin

# Automatic design usign fuzzy logic

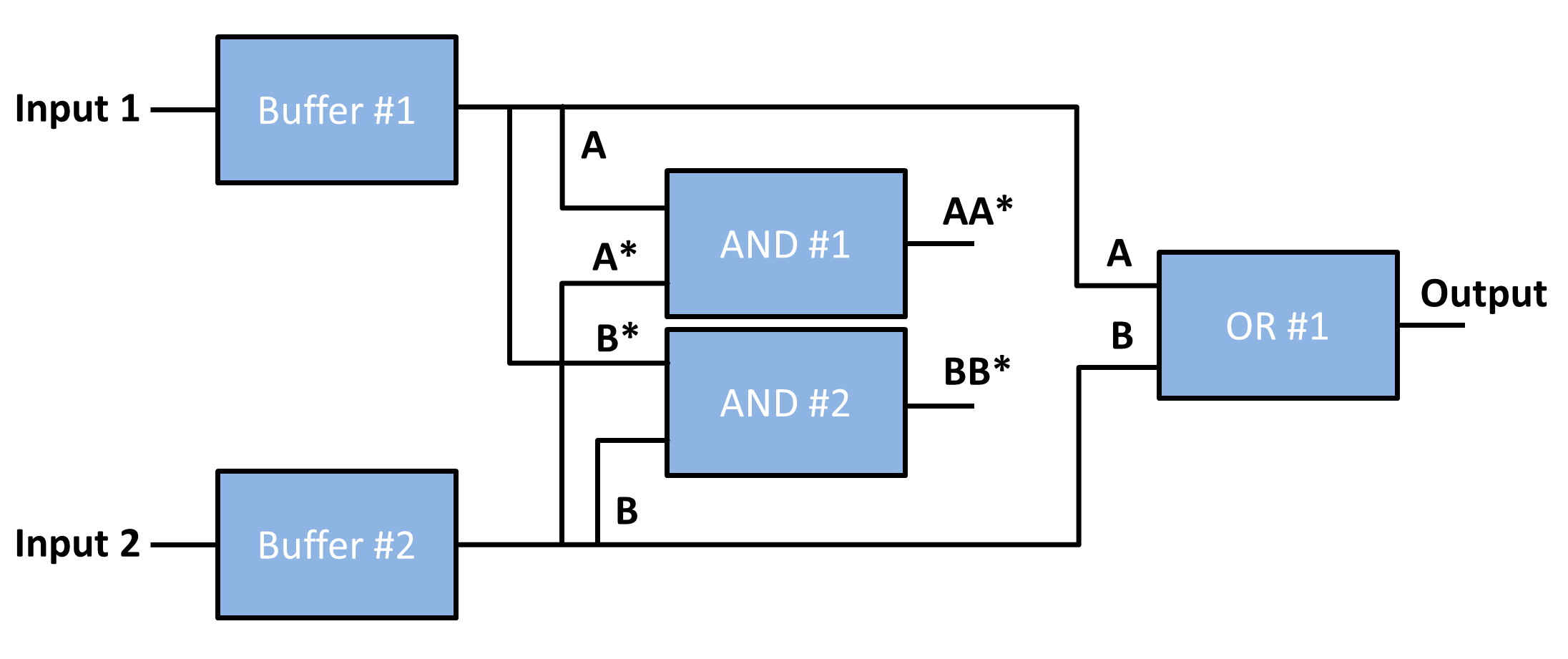
## Principle

The assumption is to work with a library of standard components that can be used to build any type of system. For each of these components, we have several models corresponding to the same logical function but with different characteristics. The various models are simulated for each component of the biosystem. The pair of components with the closest result to the intended one allows us to obtain the ideal characteristics for each mechanism involved in the system.

Thus, we created a library of logical gates containing buffers, NOT, AND, OR, NAND and NOR gates. For one-input gates such as buffers and NOT gates, we developed 5 models, and for two-inputs gates such as AND or OR gates, we developed 10 models. The selection of these models has been made by choosing mechanisms with more or less sensibility concerning inputs, more or less signal amplification and models balanced and imbalanced according to each entry. We implemented this methodology on two concrete examples: the conception of a XOR gate and of a bandpass.

## XOR gate

The first example is an XOR gate. We have already presented the application of fuzzy logic regarding the conception of this type of gate [ref], but we only studied it manually, choosing only two extreme cases for each biological mechanism. This time, with this new algorithm, we pushed this study further and simulated automatically a more important set of possible cases. This XOR gate is illustrated Fig. X.



1. Schematic of the XOR gate involving two buffers, two AND gates and one OR gate.

This XOR gate thus requires various mechanisms depicted by logical gates. First of all, two buffers for both inputs, each one synthesizing a protein and a complementary protein for the protein of the other buffer (e.g. A and B\* for input 1 buffer). Then, two AND gates are respectively used for the binding between A et A\* and B and B\*. Finally, an OR gates allows to synthesize the output protein according to A and B. This biosystem thus have every mechanism most frequently used to design biosystems, which is a good study case.

We thus have three types of gates: 2 buffers, 2 AND gates and an OR gate. To modelise the XOR gate we thus have 5\*5\*10\*10\*10 that is to say 25000 possibilities of association of these gate models.

Présentation de la méthodologie de test

Présentation des résultats de Martin sur la sélection des portes.

## Bandpass filter

# Discussion and conclusion

In this paper, we have seen the evolution of our computational core of fuzzy logic dedicated to synthetic biology. Its C-language development can be reused in another projects and also allowed us to reduce the computation time. The computation time is extremely low which also allow its use in the design step, to provide quantitative results in a step where commonly used models provided only qualitative results.

Studies on simple biosystems allowed us to highlight the need to have at least as many membership functions as cascaded stages in the system. They also showed that the increase in computation time was linear with the number of MFs used and that a good compromise between speed and accuracy of calculation is to use seven MFs.

Finally the application of the algorithm in the automatic selection of components in the design step shows interest. In a simulation time of only few seconds, we could discretize the best biological parts for the design of an XOR gate as well as for the design of a bandpass filter.

It remains to enrich the library of elementary mechanisms and to improve the automation component selection. Indeed, for the moment the script tests all possible associations between different components of a system. A choice in guiding the script would benefit the simulation time and will avoid having to simulate the set of all possible scenarios.

##### References

1. D. Endy, “Foundations for engineering biology,” Nature, vol. 438, pp.449–453, 2005.
2. J. Beal *et al*, “An End-to-End Workflow for Engineering of Biological Networks from High-Level Specifications”, ACS Synthetic Biology, vol. 1(8). 2012.
3. Canton B., Labno A., Endy D., “Refinement and standardization of synthetic biological parts and devices”, in Nature Biotechnology, vol. 26, pp. 787–793, 2008
4. Y. Gendrault, M. Madec, C. Lallement, F. Pecheux, and J. Haiech, “Synthetic biology methodology and model refinement based on micro-electronic modeling tools and languages,” Biotechnology Journal, vol. 6, pp. 796–806, 2011.
5. L.A. Zadeh, “Fuzzy Sets”, Information and Control, vol. 8, pp. 338-353, 1965.
6. J. P. Woolf and Y. Wang and J. Peter, “A Fuzzy Logic Approach to Analyzing Gene Expression Data”, Physiol. Genom. 3, pp. 9-15, 2000.
7. M. McKenna, B.M. Wilamowski, “Implementing a fuzzy system on a field programmable gate array”, International Joint Conference on Neural Networks , vol. 1, pp. 189-194, 2001.
8. H.R. Pourshaghaghi, J. Pineda de Gyvez, « Dynamic Voltage Scaling Based on Supply Current Tracking Using Fuzzy Logic Controller”, Proc. of the 16th IEEE Int. Conf. on Electronics, Circuits, and Systems, pp. 779-782, 2009.
9. G. J. Klir and B. Yuan, “Fuzzy sets and fuzzy logic”, Prentice Hall, pp. 487-499, 1995.
10. Z. Xie, L. Wroblewska, L. Prochazka, R. Weiss, and Y. Benenson, “Multi-Input RNAi-Based Logic Circuit for Identification of Specific Cancer Cells,” Science, vol. 333, pp. 1307-1311, 2011.
11. Gendrault *et al*. "Using digital electronic design flow to create a Genetic Design Automation tool" in 2012 IEEE International Conference of Engineering in Medicine and Biology Society (EMBC), San Diego (CA, USA), 28 aug.-1 sept. 2012.
12. T. Grotker, S. Liao, G.L. Martin and S. Swan, “System Design with SystemC”, ed. Kluwer Academic Publisher, 2002.