SyQUAL: a Platform for Qualitative Modelling and Simulation of Biological Systems

Rosario Distefano*, Nickolas Goncharenko†, Franco Fummi*, Rosalba Giugno*, Gary D. Bader†, Nicola Bombieri*
*Department of Computer Science, University of Verona, Verona, 37134, Italy
{rosario.distefano, franco.fummu, rosalba.giugno, nicola.bombieri}@univr.it
†The Donnelly Centre, University of Toronto, Toronto, ON M5S 3E1, Canada
{nickolas.goncharenko, gary.bader}@utoronto.ca

***Abstract*—Qualitative modelling in systems biology is increasingly adopted as it allows predicting important properties of biological systems even when quantitative information of such systems are unknown. Even though different tools for qualitative modelling have been recently proposed, their lack of automatism and their unstructured simulation core limit their applicability to non-complex biological networks. This paper presents *SyQUAL*, a platform for qualitative modelling and simulation of biological systems. It consists of two main layers: a Web-based framework that allows users to (i) import models described in the standard Systems Biology Markup Language (SBML), (ii) easily define properties to observe, and (iii) run simulations by hiding the underlying layer, that is, a SystemC-based core simulator that allows simulating the systems through a discrete event-based model of computation at different levels of details. The paper shows how *SyQUAL* has been applied to identify the attractors and to analyse the system robustness/sensitivity under perturbations of the Colitis-associated Colon Cancer (CAC) network.**

I. Introduction

Modelling and simulation of biological systems is a key requirement for integrating in-vitro and in-vivo experimental data. In-silico simulation allows testing different experimental conditions, thus helping in the discovery of the dynamics that regulate the system. These dynamics include errors in the cellular information processing that are responsible for diseases such as cancer, autoimmunity, and diabetes as well as drug effects to the system [1].

In this context, modelling approaches can be classified into two categories: *quantitative* and *qualitative* models. Quantitative modelling allows for a natural representation of molecular and gene networks and provides the most precise prediction. Nevertheless, the lack of kinetic data (and of quantitative data in general) hampers its use in many situations [2]. In contrast, qualitative models simplifies the biological reality and are often able to reproduce the system behaviour. They cannot describe actual concentration levels nor realistic time scales. As a consequence, they cannot be used to explain and predict the outcome of biological experiments that yield quantitative data. However, given a biological network consisting of input (e.g., receptors), intermediate, and output (e.g., transcription factors) signals, they allow studying the input-output relationships through discrete simulations [3].

In the last decade, different qualitative approaches have been successfully used to extrapolate insights of system networks. Nevertheless, they have shown having two main limitations: (i) they do not support the simulation complexity of large networks, and (ii) they lack of automation in analysing biological properties such as complex attractors, molecule vulnerability, and dose response [2].

This paper presents *SyQUAL*, a platform for qualitative modelling and simulation of biological systems. Differently from all the tools in literature, *SyQUAL* allows performing both automatic and efficient system simulation. Being based on languages and design tools well-established in the *Electronic Design Automation* (EDA) field, it allows addressing high computational costs normally associated to the modelling and simulation of biological systems [4], [5]. The simulation core relies on a discrete event-based framework developed in SystemC, which is the de-facto reference standard language in EDA for efficient and accurate simulations of systems at different levels of abstraction.

The platform provides both synchronous and asynchronous updating methods for simulation, where the asynchronous method relies on a time-delayed updating scheme controlled by topology-based constraints. A different time delay τ_i is assigned to each link between two nodes, and each node is evaluated according to a specific timescale. Biological events are characterized by different timescales. For example, expression of genes is not an instantaneous process; infact, a hypothetical biochemical reaction can take from milliseconds up to few seconds.This allows providing more realistic network simulations by avoiding unrealistic node updates. *SyQUAL* fully supports both Systems Biology Markup Language (SBML) level 2 and SBML level 3-*qual* network descriptions, thus extending its portability to all the system networks described in such a standard language.

The paper shows and compares the results obtained by applying *SyQUAL* and the most representative qualitative tools at the state of the art to identify the attractors and to analyse the system robustness/sensitivity under perturbations of the Colitis-associated Colon Cancer (CAC) network.

The paper is organized as follows. Section II presents some background and the state of the art. Section III presents the *SyQUAL* platform in detail. Section IV presents the experimental results, while Section V is devoted to the concluding remarks.

II. Background and state of the art

Proposed by Kauffman [6] and Thomas [7], the discrete logic-based dynamical models have been successfully applied for modeling biological systems, such as the cell cycle [8], the gene regulatory system [9], and signalling networks [10].

Several software and tools have been developed to address the logic modelling of biological systems. *BoolNet* [11], *SimBoolNet* [12], *GINsim* [13], *ADAM* [14], *The Cell Collective* [15], and *CellNetAnalyzer* [16] are representative modelling environments, many of them extended with the concept of *multi-valued* boolean dynamic modelling.

Piecewise linear differential equations-based frameworks, such as *BooleanNet* [17], represent an alternative way to model the boolean network response in a continuous manner, by converting boolean functions to their continuous counterparts. In *ChemChains* [18], modelling of discete models relies on discrete active/inactive ratio-based simulations, which allow users to interact with the model in a continuous domain. In contrast, logic-based ordinary differential equations (ODEs) frameworks, such as *Odefy* [19] and *SQUAD* [20], transform a boolean model in a corresponding network topology-derived system of continues differential equations, without requiring detailed kinetic information.

All these software and tools can be categorised on the basis of their supported updating methods. The *asynchronous* updating method is the distinguishing key, since the synchronous one adopts the same global updating strategy. *Random Order Asynchronous* (ROA) [21], *General Asynchronous* (GA) [21], [22], [23], *Priority Class* [24], [8], and *Ranked Asynchronous* (RA) [22] are the most common asynchronous methods. Considering a network of N nodes:

1) Random Order Asynchronous (ROA): All nodes are updated at the same time step, but in a random order, such that no node is updated twice in the same time step. In the updating step, a random permutation $Q = Q_1, \ldots, Q_N$ is generated from the ordered set $\{1, \ldots, N\}$. Then, the state of node i, the Q_ith element of Q, at time $t+1$ is calculated as follows:

$$X_i(t+1) = F_i(X_1(t_{1,i}), \ldots, X_N(t_{N,i})) \quad \forall\, i = 1, .., N$$

where F_i is the boolean function that describes the state of node i at time $t+1$, X_i is the state of the node i at a specific time, and

$$t_{j,i} = \begin{cases} t & \text{if } Q_j > Q_i \\ t+1 & \text{if } Q_j < Q_i \end{cases}.$$

This means that if the input node j has been updated at the $(t+1)$th time step, then $X_j(t+1)$ should be used in the right hand side of the equation. If an input node has not been updated (e.g., the last update was in the tth time step), then $X_j(t)$ should be used in the right hand side of the equation.

2) General Asynchronous (GA): In this method, a randomly selected node is updated at each time step. In the updating step, a random element of the ordered set $\{1, \ldots, N\}$, i is selected. Then, the state of node i, at time $t+1$ is calculated as follows:

$$X_i(t+1) = F_i(X_1(t), \ldots, X_N(t))$$

where F_i is the boolean function describing the state of node i at time $t+1$, X_j is the state of a node at a given time step. Note that only node (i) is updated at a given time (i.e., this could lead to update the same node multiple times in a row).

3) Priority Class (PA): The nodes are updated either synchronously or asynchronously (see *GA*) in a specific order. Each node belongs to one of the different priority classes $C_1, C_2, ..., C_p$, with $p \leq N$. Each class C_i has both a rank and a chosen updating method (synchronous or asynchronous). In the updating class, nodes with the highest ranked priority class are updated first, and are updated with the updating method choosen for the class. Classes of the same rank are updated independently and asynchronously and classes of lower rank occur after the highest ranked classes.

4) Ranked Asynchronous (RA): It shares the same approach of the *Priority Class*. However, a *Ranked Asynchronous* method adopts only the asynchronous updating method.

Each of the asynchronous method listed above has some limitation. Methods such as *GA* and *ROA*, can lead to the indiscriminate enumeration of all possible sequences of the node updating, which includes many incompatible or unrealistic pathways. This leads to *biologically implausible* simulations of the qualitative networks. In contrast, methods of the PA or RA classes are more realistic but, on the other hand, they rely on an a-priori knowledge (which is not always available) to categorise the network nodes in classes.

Less precise but easier to adopt are the methods with synchronous updating scheme like *The Cell Collective* [15] or *CellNetAnalyzer* [16]. This last has been extended to the semi-quantitative domain through the integration of *Odefy* [19].

Another important limitation of some of the analysed tools is the input data format. *The Cell Collective*, *ChemChains*, and *CellNetAnalyser* require proprietary formats, without providing any support to biological systems described in SBML.

III. The *SyQUAL* Platform

SyQUAL provides a whole serie of facilities (i) to model and simulate a given biological system described through an SBML model, and (ii) to perform automatised experiments under specific and controlled conditions. Since *SyQUAL* is a web-oriented platform, it does not require any particular library or adjustment to be used, thus allowing the user to focus only on experiment setting and execution.
The platform is organised into two main blocks (see Figure 1): The front-end, which implements the web-user interface (UI), and the back-end, which implements a SBML validator and a SBML-to-SystemC translator.

A. Front-end block

The *SyQUAL* interface relies on open source languages, such as Javascript and HTML5, and on free libraries, such as JQuery, Vis.js, and Bootstrap 3, in order to create a fully-responsive cross-browser application. The main modules are:

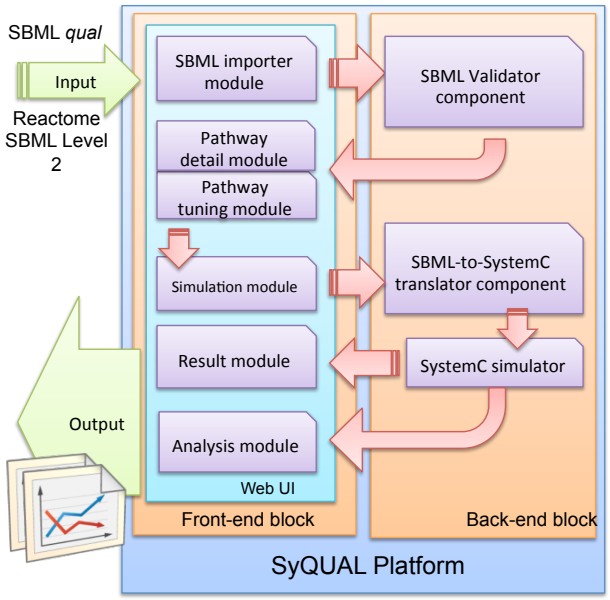

Fig. 1. Overview of the *SyQUAL* platform.

1) SBML importer module: It represents the starting point for any new user-defined project. It has been designed to facilitate the uploading of SBML-based biological models, such as those provided by the *Reactome* database [25]. This module also provides an additional way to increase the amount of information to be associated to the imported SBML description. As an exmaple, the user can specify a list of elements and their corresponding *Uniprot IDs*, in order to identify potential drug targets for the robustness/sensitivity analysis. This represents a useful features, especially when the SBML description lacks of these details.

2) Pathway detail module: Through this module, part of the information extracted during the SBML importing process is shown to the user for a check. *SyQUAL* reports the complete list of biological system elements (e.g., genes, proteins, miRNAs) supplied by their Uniprot IDs if described in the SBML model, and a list of DrugBank drugs associated to each biological element if a correspondence between the UniprotID and one or more drugs exists. DrugBank drugs are provided into seven different drug categories, which include approved drugs (according to the "Food and Drug Administration" - FDA [26]), small molecular drugs, experimental drugs, investigation drugs, illicit drugs, withdraw drugs, and biotech drugs.

Each drug reports, if available, a comprehensive set of information, such as, the pharmacological actions (i.e., yes, no, unknown), the action (i.e., inhibitor, activator, etc.), and the drug ID/name.

Information about drugs can be used to identify potential targets. Depending on the drug biological action, *SyQUAL* alters the system behaviour for the sensitivity analysis. Moreover, since drugs are divided into categories, *SyQUAL* helps the user to perform drug-dependent experiments.

3) Pathway tuning module: It provides a set of facilities to customise the simulation environment. For a given simulation to be performed, it is possible to:

- Select a set of stimuli to be activated. Stimuli are system elements that act as starting points for the simulation. Each stimulus can be (i) activated only during the initial part of a simulation, or (ii) kept always activated during the whole simulation.
- Select a set of drugs to be used, distinguishing them by their pharmacological action and effect.
- Select a set of system elements to be knocked down. In this case, the user can force the suppression of one or more elements, thus simulating the presence of hypothetical drugs, even if no drug is provided by *SyQUAL*.

In addition, *SyQUAL* provides a graphical network representation of the biological system to show the network topology.

4) Simulation module: This module applies the specified customisation to settle and perform specific simulations. This allows *SyQUAL* to simulate the system network by altering the system behaviour in different ways:

- Unsupervised mode: *SyQUAL* automatically performs the simulations according to the following options:
  -) *Systematic knocking down of each system element (SyK).* It performs $|N|$ distinct simulations (where N represents the number of system elements) in which each system element is knocked down at a time.
  -) *Drug-driven knocking down of system element targets (DDK).* It performs $|K| \leq |N|$ distinct simulations, where K represents the number of distinct *set* of system elements knocked down at a time as an effect of a specific drug. *SyQUAL* allows distinguishing two main cases: A *single drug* that targets more than one system elements (in this case, a single simulation knocks down all drug target system elements), and *multiple drugs* that target a single system element (i.e., a single simulation knocks downs only the targeted system element). This allows *SyQUAL* to avoid repeating the same simulations, thus reducing the computational time.
  -) *Drug-driven knocking down of system element target pair combination (DDKp).* It applies the concept of *synthetic lethality* [27], which appears when a combination of mutations in two or more genes leads to cell death, whereas a mutation in only one of these genes does not, and by itself is said to be viable. Synthetic lethality indicates functional relationships between genes. *SyQUAL* allows knocking down a set of system elements according to a list of drug targets or to a list of system elements manually selected.
- Supervised mode: *SyQUAL* performs a single simulation according to one of the following options:
  -) *User-driven knocking down of system elements (UDK).* It allows knocking down a set of system elements by manually selecting a list of drugs to be used or a list of system elements to be knocked down. With this option, many system elements at a time can be knocked down in

a single simulation.

-) *User-driven knocking down of system elements under multiple stimuli (UDKm)*. Similarly to *UDK*, it performs *multiple* simulations based on the same set of system elements knocked down. However, by specifying a list of stimuli, the *UDKm* option combines all selected stimuli, performing *M* simulations with the same stimuli combinations:

$$M = \sum_{k=1}^{h=n} C_n^k \tag{1}$$

where *n* represents the number of selected stimuli.

5) Result module: It shows the simulation results according to a specific selected option and updating method. Since an experiment is a collection of distinct simulations, *SyQUAL* provides:

- A heatmap reporting the activity levels (i.e., the time frame in which a specific system element is active) over the time of all system elements under normal conditions.
- For each simulation, a heatmap reporting the activity levels over the time regarding only system elements that have not been knocked down.
- The set of identified attractors.
- A comprehensive expression profile heatmap of the elements, which is created by merging all activity levels and clustered according to the Pearsons correlation.

6) Analysis module: It provides a way to examine the activity levels of the system elements by using a normalised (in the range [0,1]) numerical representation. The activity levels can be analysed by specifying the performed option and updating method as well as the set of system elements of interest. This allows, without any graphical representation, investigating and quantifying how the the activity level of elements level influence each other.

B. Back-end block

It consists of three main modules.

1) SBML Validator: This module plays a key role to retrieve information from a given biological system described in SBML. *SyQUAL* supports the SBML level 2 (e.g., models provided by the *Reactome* database), and the SBML level 3 *qual*. By supporting the SBML level 2, *SyQUAL* generates a fully-comprehensive qualitative reaction boolean network, keeping important details related to the molecular interaction nature, such as inhibition, stimulation, catalysis. In contrast, the SBML *qual* preserves a subset of these details, thus limiting the number and the quality of observable behaviours for a faster simulation. As a de-facto reference database of biological systems, Reactome provides SBML level 2 models with an SBML annotation that describes all involved elements, reactions, and reactants, and sometimes provides an SBGN (*Systems Biology Graphical Notation* [28]) annotation describing the system through a graphical representation.

SyQUAL creates a fully-comprehensive network generated by merging the SBML and SBGN annotations, when the last is available. To accomplish with this task, *SyQUAL* relies on the *LibSBML API* (the Python Library version) to acquire and manipulate the SBML model.

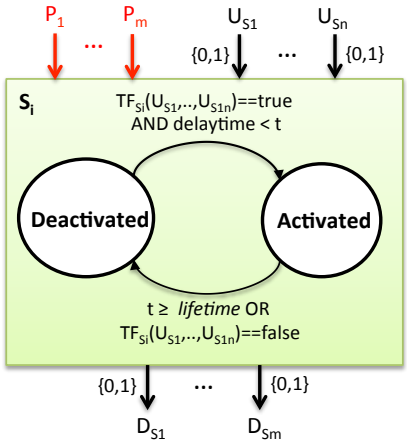

Fig. 2. The *Finite State Machine* representation of the system elements.

2) SBML-to-SystemC Mapper: Biological elements and interactions are translated into SystemC processes and signals. Processes are the central building blocks in a SystemC description. A SystemC system description can be seen as a set of concurrent processes that communicate each other using clock-dependent signals.

SyQUAL maps each aspect of the SBML-based biological element as follow:

- The element behaviour is modelled through a *Finite State Machine* (FSM) [29]. The FSM is used to formally model the system element through a boolean representation to manage the element state (e.g. activated, deactivated), the state transitions, and the guard conditions.
  Figure 2 shows the template defined to represent each element through a FSM. It consists of the following input and output signals:
  -) *Parameters (P)*: these inputs values are unknown at modelling time, and depend on the environment characteristics. Examples are the *delay time* (i.e., the time spent by a biological element to encounter its target), and the *lifetime* (i.e., the maximum time after activation, in which the biological element carries out its biological function). *SyQUAL* performs the *parametrization* (i.e., assignment of parameter values) during simulation.
  -) *Upstream inputs (U_s)*: these topologically-depended input are generated at simulation time, as the result of the interaction with their upstream biological elements. The *TF* function represents the element *transfer function*, a boolean function that depends on the current values of U_s, and it is used to evaluate the element activation/deactivation.
  -) *Downstream outputs (D_s)*: the output values are generated at simulation time and depend on the role of the biological element (i.e., the element can acts as activator and/or inhibitor for its downstream elements).

SyQUAL implements the FSM model of each biological element (i.e., genes, proteins) through a SystemC process, which is sensible to events coming as input signals. Any new event (i.e., signal value variation) that occurs on a specific element input, leads to a new evaluation of its guard conditions, and to a corresponding updating of its state and output signals. The *SyQUAL* simulation core relies on the discrete event-based kernel of the SystemC simulator, which is optimized to provide efficient simulations of complex networks.

The system simulation can be performed by selecting one of the following updating methods:

- Synchronous updating method. It represents the simplest and most computational efficient method. Given a boolean representation of a biological network, each element is evaluated by using a global clock. The time step (which we call *delaytime*) is equal for all system elements. All interactions between system elements are performed in a synchronous way. For each time step, all nodes status are evaluated, according to their own boolean rules (logic input combination). This method provides the best way to investigate at high-level of abstraction some basic behaviours, such as feedback loops, particular signals paths, and attractors.
- Asynchronous updating method. It provides a different delay time for each system element. All interactions between system elements are performed in an asynchronous way. It relies on the concept of *lifetime*, which represents the maximum time in which an element can execute its biological function. This method allows investigating more accurately (i.e., at low level of abstraction) the system behaviours.

IV. Experimental results

In order to better explain how *SyQUAL* can be applied for modelling and simulation of biological systems, we analysed, as case study, the *Colitis-associated Colon Cancer* (CAC) [30]. CAC is a complex enough network studied both in-silico and in-vitro to understand dynamics behind inflammation-associated tumourigenesis and to identify potential novel therapies. We refer to the work in [30] to compare our simulation results with the in-vitro experimental results. The work provides insights to understand how molecular mechanisms lead to Colitis-associated Colon Cancer and a refined Boolean Network model related to the growth and survival of preneoplastic epithelial cells. According to [30], the system network can be decomposed in two main parts: The *IEC* part, which contains elements related to the intracellular signalling, and a second part associated to the immune microenvironment, which contains elements such as immune cells, cytokines and chemokines. The analysis and the experimental observations have been conducted according to four main micro-environments (conditions), a given set of input stimuli, and a list of nodes that have been systematically knocked down as reported in Table I. A distinct micro-environment is associated to:

TABLE I
List of main microenvironments.

Condition	Initially On	Initially Off	Fixed On	Fixed Off
Non-inflammatory microenvironment		Prolifetation, Apoptosis	APC	IL6, IL12, IL4, TH1, TH2, IL10, TREG, IFNG, MAC, CCL2, TGFB, CTL, TNFA, PGE2, DC
Normal inflammation response	DC	Prolifetation, Apoptosis	APC	IL6, IL12, IL4, TH1, TH2, IL10, TREG, IFNG, MAC, CCL2, TGFB, CTL, TNFA, PGE2
Pro-tumor microenvironment		Prolifetation, Apoptosis	DC	
Pro-tumor microenvironment and P53 inactivation		Prolifetation, Apoptosis	DC	P53

- A set of activated nodes (stimuli) at the beginning of the simulation (*Initially On*).
- A set of deactivated nodes at the beginning of the simulation (*Initially Off*).
- A set of nodes (stimuli) always kept active during the whole simulation (*Fixed On*).
- A set of nodes always kept deactivated during the whole simulation (*Fixed Off*).

We investigated attractors dynamics, both fixed-point (stable state) and cyclic (with regularly recurring states), to understand whether *SyQUAL* was able to reproduce experimental observations associated to each specific micro-environment.

The attractor analysis plays an important role to identify system dynamics, since they are the corresponding phenotypes for a biological system. As a direct consequence of such an analysis, we tested the correlation among the network nodes to identity potential inaccurate behaviours. An example is given by the Th1 and Th2 responses. The immune microenvironment influences the epithelial cell growth and survival through (i) the releasing of cytokines or (ii) the direct interactions between epithelial cells and immune cells. In this context, Th1 and Th2 responses counteract each other as follows:

- *Th1*. Cellular immune system. Maximizes the killing efficacy of the macrophages.
- *Th2*. Humoral immune system. Stimulates B-cells into proliferation, and increases the neutralisation of antibody production.

We performed the attractor analysis of the case study with *SyQUAL* and with the most representative tools for qualitative modelling and simulation of biological systems in literature. We selected *BoolNet*, *SimpleBool*, *BooleanNet*, and *GINsim* since they share common features, such as, the support for SBML *qual* input models and the support of both synchronous and asynchronous updating scheme during simulation. Moreover, relying on the *GA* and *ROA* updating methods, these tools do not require any prior information, in contrast with the *PA* and *RA* updating methods. The comparison has been executed on different machines. *BoolNet*, *SimpleBool*, *BooleanNet*, and

GINsim can be installed standalone, while *SyQUAL* requires a specific environment because of its web-based nature. However, all machines are characterized by similar technical features.

We run the system analysis through synchronous and asynchronous simulations, by testing all the conditions reported in Table I, that is, *non-inflammatory microenvironment* (1), *normal inflammation response* (2), *pro-tumor microenvironment* (3), *pro-tumor microenvironment and P53 inactivation* (4).

Table II reports the results obtained with the synchronous updating scheme in terms of simulation time, the corresponding standard deviation (for a batch of one hundred executions), and the number of identified attractors.

The table underlines that all the tools (except for *GINsim* that does not support the complexity of the case study) led to the identification of the same number and type of attractors. This was expected since the tools are all based on the same simple synchronous updating method. However, differently from the other tools and correctly (according to the results in [30]), *SyQUAL* identified that the attractor associated to the *normal inflammation response* also includes the dendritic cell (*DC*) activation. This result is motivated by the fact that *DC* depends on *CCL2*, *TNFA*, and *IL10*. Since the *normal inflammation response* micro-environment requires *CCL2*, *TNFA*, and *IL10* to be knocked down, a hypothetical activation of *DC* must keep it active during the whole simulation.

TABLE II
EXPERIMENTAL RESULTS WITH SYNCHRONOUS SIMULATIONS

Tool	Cond.	Timing in ms		#attract.
		exec. time	sd	
BoolNet	1	1.16	0.51	1
	2	1.02	0.47	1
	3	1.10	0.56	6
	4	1.22	0.66	6
SimpleBool	1	424.15	30.03	1
	2	399.47	12.48	1
	3	392.75	17.04	6
	4	428.49	48.40	6
GINsim	1	3,000.00	1,000.00	1
	2	3,000.00	1,000.00	1
	3	Out of memory		
	4	Out of memory		
BooleanNet	1	11.59	1.64	1
	2	11.58	1.64	1
	3	15.07	0.23	6
	4	16.340	4.79	6
SyQUAL	1	11.46	1.69	1
	2	10.97	1.56	1
	3	13.03	2.04	6
	4	12.21	1.87	6

Table III shows the results obtained with the system simulation based on asynchronous updating schemes. The results underline that such an accurate simulation leads to an increasing of the execution time from one to three orders of magnitude with the tools in literature. With this simulation accuracy, also *SimpleBool* does not support the case study complexity by leading to no attractors found (in [30], the network has been simplified from 70 to 28 nodes for the attractor analysis). Table III underlines that only *BooleanNet* and *SyQUAL* can correctly lead to a greater number of attractors, thus underlying the sensitivity of the network to the element parametrization. Nevertheless, *BooleanNet* pays such an accuracy with a prohibitive price in terms of execution time.

Finally, we underline that we conducted the comparison of the tools by adopting only the user-driven knocking down of system elements (UDK) since it is the only modality implemented by the tools in literature to alter the system. Such a supervised mode, when adopting the tools in literature, required a strong intervention by the user in the code to knock down the system elements. In contrast, thanks to the front-end block, *SyQUAL* allowed setting the simulations and reporting the results instantly with the Web UI support.

TABLE III
EXPERIMENTAL RESULTS WITH ASYNCHRONOUS SIMULATIONS

Tool	Cond.	Updating	Timing in ms		#attract.
			exec. time	sd	
BoolNet	1	GA	5.45	1.37	1
	2	GA	5.02	0.80	1
	3	GA	15.41	31.73	6
	4	GA	4.67	0.95	6
SimpleBool	1	GA	10,424.99	559.21	-
	2	GA	10,160.82	190.80	-
	3	GA	10,135.45	99.81	-
	4	GA	10,138.06	139.48	-
	1	ROA	37,867.58	325.07	-
	2	ROA	43,864.31	1,489.34	-
	3	ROA	46,682.54	732.65	-
	4	ROA	46,016.95	1,119.86	-
GINsim	1	GA	3,000.00	1,000.00	1
	2	GA	3,000.00	1,000.00	1
	3	GA	Out of memory		
	4	GA	Out of memory		
BooleanNet	1	GA	11,499.89	493.73	1
	2	GA	10,560.72	207.89	1
	3	GA	12,023.21	227.93	6
	4	GA	11,957.15	216.84	6
SyQUAL	1	Time-delayed	10.43	1.87	3
	2	Time-delayed	11.60	1.92	2
	3	Time-delayed	12.38	1.49	5
	4	Time-delayed	13.82	1.71	5

V. CONCLUSIONS

This work proposed *SyQUAL*, a platform for qualitative modelling and simulation of biological systems. The paper showed and compared the results obtained by applying SyQUAL and the most representative qualitative tools at the state of the art to identify the attractors and to analyse the system robustness/sensitivity under perturbations of the Colitis-associated Colon Cancer (CAC) network. The results underlined the best trade-off between accuracy of results and simulation performance with regard to the tools in literature.

REFERENCES

[1] E. Gonalves, J. Bucher, A. Ryll, J. Niklas, K. Mauch, S. Klamt, M. Rocha, and J. Saez-Rodriguez, "Bridging the layers: Towards integration of signal transduction, regulation and metabolism into mathematical models," *Molecular BioSystems*, vol. 9, no. 7, pp. 1576–1583, 2013.

[2] N. Le Novere, "Quantitative and logic modelling of molecular and gene networks," *Nat Rev Genet*, vol. 16, no. 3, pp. 146–158, Mar. 2015. [Online]. Available: http://dx.doi.org/10.1038/nrg3885

[3] R. Samaga and S. Klamt, "Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks," *Cell Communication and Signaling*, vol. 11, no. 1, 2013. [Online]. Available: http://dx.doi.org/10.1186/1478-811X-11-43

[4] N. Bombieri, R. Distefano, G. Scardoni, F. Fummi, C. Laudanna, and R. Giugno, "Dynamic modeling and simulation of leukocyte integrin activation through an electronic design automation framework," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8859, pp. 143–154, 2014.

[5] R. Distefano, F. Fummi, C. Laudanna, N. Bombieri, and R. Giugno, "A systemc platform for signal transduction modelling and simulation in systems biology," in *Proceedings of the 25th Edition on Great Lakes Symposium on VLSI*, ser. GLSVLSI '15, 2015, pp. 233–236.

[6] S. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, pp. 177–178, 1969.

[7] R. Thomas, "Boolean formalization of genetic control circuits," *Journal of theoretical biology*, vol. 42, no. 3, pp. 563–585, 1973.

[8] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle," *Bioinformatics*, vol. 22, no. 14, pp. e124–e131, 2006.

[9] C. E. Giacomantonio and G. J. Goodhill, "A boolean model of the gene regulatory network underlying mammalian cortical area development," *PLoS Comput Biol*, vol. 6, no. 9, p. e1000936, 2010.

[10] R. Schlatter, K. Schmich, I. A. Vizcarra, P. Scheurich, T. Sauter, C. Borner, M. Ederer, I. Merfort, and O. Sawodny, "On/off and beyond-a boolean model of apoptosis," *PLoS Comput Biol*, vol. 5, no. 12, p. e1000595, 2009.

[11] C. Müssel, M. Hopfensitz, and H. A. Kestler, "Boolnet: an r package for generation, reconstruction and analysis of boolean networks," *Bioinformatics*, vol. 26, no. 10, pp. 1378–1380, 2010.

[12] J. Zheng, D. Zhang, P. F. Przytycki, R. Zielinski, J. Capala, and T. M. Przytycka, "Simboolnet: a cytoscape plugin for dynamic simulation of signaling networks," *Bioinformatics*, vol. 26, no. 1, pp. 141–142, 2010.

[13] A. Naldi, D. Berenguier, A. Fauré, F. Lopez, D. Thieffry, and C. Chaouiya, "Logical modelling of regulatory networks with ginsim 2.3," *Biosystems*, vol. 97, no. 2, pp. 134–139, 2009.

[14] F. Hinkelmann, M. Brandon, B. Guang, R. McNeill, G. Blekherman, A. Veliz-Cuba, and R. Laubenbacher, "Adam: analysis of discrete models of biological systems using computer algebra," *BMC bioinformatics*, vol. 12, no. 1, p. 1, 2011.

[15] T. Helikar, B. Kowal, S. McClenathan, M. Bruckner, T. Rowley, A. Madrahimov, B. Wicks, M. Shrestha, K. Limbu, and J. A. Rogers, "The cell collective: toward an open and collaborative approach to systems biology," *BMC systems biology*, vol. 6, no. 1, p. 96, 2012.

[16] S. Klamt, J. Saez-Rodriguez, and E. D. Gilles, "Structural and functional analysis of cellular networks with cellnetanalyzer," *BMC systems biology*, vol. 1, no. 1, p. 1, 2007.

[17] I. Albert, J. Thakar, S. Li, R. Zhang, and R. Albert, "Boolean network simulations for life scientists," *Source code for biology and medicine*, vol. 3, no. 1, p. 1, 2008.

[18] T. Helikar and J. A. Rogers, "Chemchains: a platform for simulation and analysis of biochemical networks aimed to laboratory scientists," *BMC systems biology*, vol. 3, no. 1, p. 1, 2009.

[19] J. Krumsiek, S. Pölsterl, D. M. Wittmann, and F. J. Theis, "Odefy-from discrete to continuous models," *BMC bioinformatics*, vol. 11, no. 1, p. 1, 2010.

[20] A. Di Cara, A. Garg, G. De Micheli, I. Xenarios, and L. Mendoza, "Dynamic simulation of regulatory networks using squad," *BMC bioinformatics*, vol. 8, no. 1, p. 1, 2007.

[21] I. Harvey and T. Bossomaier, "Time out of joint: Attractors in asynchronous random boolean networks," in *Proc. of European Conference on Artificial Life*. MIT Press, Cambridge, 1997, pp. 67–75.

[22] M. Chaves, R. Albert, and E. D. Sontag, "Robustness and fragility of boolean models for genetic regulatory networks," *Journal of theoretical biology*, vol. 235, no. 3, pp. 431–449, 2005.

[23] A. Saadatpour, I. Albert, and R. Albert, "Attractor analysis of asynchronous boolean models of signal transduction networks," *Journal of theoretical biology*, vol. 266, no. 4, pp. 641–656, 2010.

[24] R. Thomas, "Regulatory networks seen as asynchronous automata: a logical description," *Journal of theoretical biology*, vol. 153, no. 1, pp. 1–23, 1991.

[25] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, L. Matthews *et al.*, "Reactome: a knowledgebase of biological pathways," *Nucleic acids research*, vol. 33, no. suppl 1, pp. D428–D432, 2005.

[26] "(1996) Guidance for industry: good clinical practice: consolidated guidance. Rockville (Maryland): U.S. Department of Health, Education, and Welfare. Food and Drug Administration." http://www.fda.gov/cder/guidance/959fnl.pdf, accessed: 2007-10-27.

[27] L. J. Albright and E. M. Wilson, "Sub-lethal effects of several metallic saltsorganic compounds combinations upon the heterotrophic microflora of a natural water," *Water Research*, vol. 8, no. 2, pp. 101–105, 1974.

[28] N. Le Novere, M. Hucka, H. Mi, S. Moodie, F. Schreiber, A. Sorokin, E. Demir, K. Wegner, M. I. Aladjem, S. M. Wimalaratne *et al.*, "The systems biology graphical notation," *Nature biotechnology*, vol. 27, no. 8, pp. 735–741, 2009.

[29] D. Coati, R. Distefano, N. Bombieri, F. Fummi, M. Mirenda, C. Laudanna, and R. Giugno, "A systemc-based platform for assertion-based verification and mutation analysis in systems biology," in *LATS 2016 - 17th IEEE Latin-American Test Symposium*, 2016, pp. 159–164.

[30] J. Lu, H. Zeng, Z. Liang, L. Chen, L. Zhang, H. Zhang, H. Liu, H. Jiang, B. Shen, M. Huang *et al.*, "Network modelling reveals the mechanism underlying colitis-associated colon cancer and identifies novel combinatorial anti-cancer targets," *Scientific reports*, vol. 5, 2015.