

SOFTWARE

Open Access



# CRA toolbox: software package for conditional robustness analysis of cancer systems biology models in MATLAB

Fortunato Bianconi<sup>1\*</sup>, Chiara Antonini<sup>2†</sup>, Lorenzo Tomassoni<sup>2†</sup> and Paolo Valigi<sup>2</sup>

## Abstract

**Background:** In cancer research, robustness of a complex biochemical network is one of the most relevant properties to investigate for the development of novel targeted therapies. In cancer systems biology, biological networks are typically modeled through Ordinary Differential Equation (ODE) models. Hence, robustness analysis consists in quantifying how much the temporal behavior of a specific node is influenced by the perturbation of model parameters. The Conditional Robustness Algorithm (CRA) is a valuable methodology to perform robustness analysis on a selected output variable, representative of the proliferation activity of cancer disease.

**Results:** Here we introduce our new freely downloadable software, the CRA Toolbox. The CRA Toolbox is an Object-Oriented MATLAB package which implements the features of CRA for ODE models. It offers the users the ability to import a mathematical model in Systems Biology Markup Language (SBML), to perturb the model parameter space and to choose the reference node for the robustness analysis. The CRA Toolbox allows the users to visualize and save all the generated results through a user-friendly Graphical User Interface (GUI). The CRA Toolbox has a modular and flexible architecture since it is designed according to some engineering design patterns. This tool has been successfully applied in three nonlinear ODE models: the Prostate-specific *Pten*<sup>-/-</sup> mouse model, the Pulse Generator Network and the EGFR-IGF1R pathway.

**Conclusions:** The CRA Toolbox for MATLAB is an open-source tool implementing the CRA to perform conditional robustness analysis. With its unique set of functions, the CRA Toolbox is a remarkable software for the topological study of biological networks. The source and example code and the corresponding documentation are freely available at the web site: <http://gitlab.ict4life.com/SysBiOThe/CRA-Matlab>.

**Keywords:** Ordinary differential equation models, Conditional robustness analysis, MATLAB package, Signaling networks

## Background

In Systems Biology, mathematical modeling and computational software are important tools to unravel the complexity of biological systems and predict their behavior under different perturbations [1]. Typically, many models consist of a set of Ordinary Differential Equations (ODEs) which allow understanding and reproducing the dynamic behavior of molecular interactions through simulations and integration of the ODEs [2]. To support mathematical

modeling of biological networks, the use of software tools has grown substantially in recent years. These software are designed to assist the users at different stages of the modeling process, from model generation to parameter estimation and model analysis.

In cancer research, the use of systems biology approaches is particularly useful to elucidate mechanisms of tumorigenesis and tumor resistance. Computational predictive models, integrated with patient data, help scientists in the validation of new and durable therapies [3]. In order to discover effective and targeted drugs, robustness is one of the most relevant properties of cancer signaling networks to investigate. Robustness is defined as

\*Correspondence: [fortunato.bianconi@gmail.com](mailto:fortunato.bianconi@gmail.com)

†Chiara Antonini and Lorenzo Tomassoni contributed equally to this work.

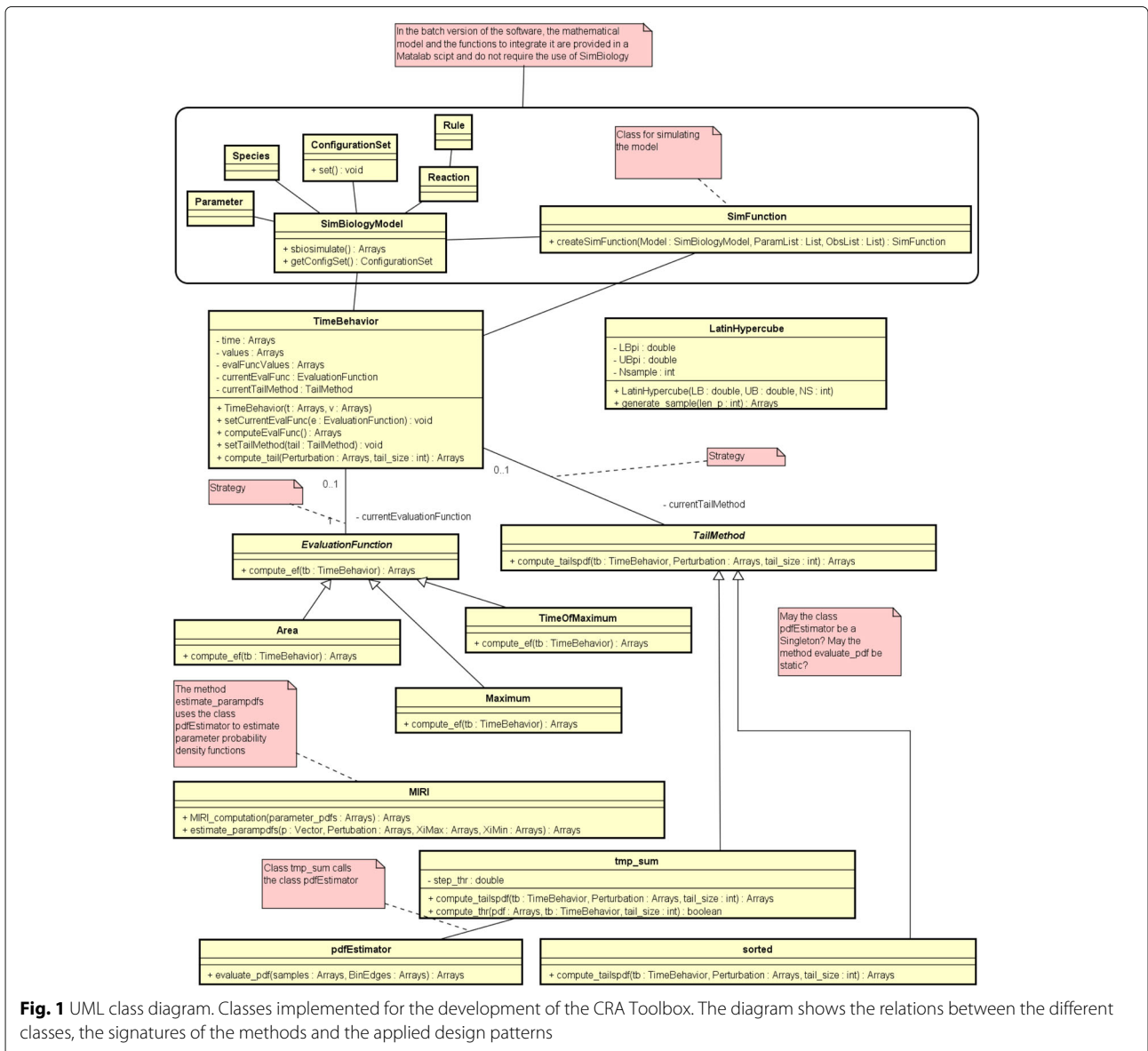
<sup>1</sup>Independent Researcher, Belvedere 44, 06036 Montefalco, Perugia, Italy  
Full list of author information is available at the end of the article



the ability of a biological system to maintain its functionalities against internal and external perturbations [4]. In more detail, cancer robustness is a quantitative measure of the tumor proliferation attitude against extracellular inputs. Thus, understanding new ways to reduce robustness of the cell proliferation activity is a key issue in cancer systems biology. Since cell growth is driven by protein interaction networks, the proliferation activity can be quantified by looking at the activation of a protein involved in the proliferation process [5]. In mathematical modeling, this can be done by perturbing model parameters and analyze how the concentration of the protein of interest changes over time. An algorithm developed for this purpose is the Conditional Robustness Algorithm

(CRA) proposed in [5]. This algorithm, through computational perturbations and simulations, identifies a small number of nodes in the cancer network which influences most the activity of the proliferation indicator. As a result, by conditioning these nodes with specific drugs, it may be possible to reduce the tumor robustness.

Robustness of mathematical model problem is well studied in literature except when the models are based on nonlinear ODE. A class of methods is aimed at analyzing the geometry and the volume of the feasible region, which is the region in the parameter space that allows the system to properly work [6]. Moreover, there are other algorithms that infer the robustness of a model looking at its topology, such as the number and the structure of positive and



**Fig. 1** UML class diagram. Classes implemented for the development of the CRA Toolbox. The diagram shows the relations between the different classes, the signatures of the methods and the applied design patterns

negative loops [7]. However, these techniques are typically applied to mathematical models whose parameters are not kinetic. Another and different category of methods is the Global Sensitivity Analysis (GSA) class of algorithms [8]. They are similar to CRA and are involved in the analysis of the uncertainties in kinetic model parameters through the sampling of the parameter space. Despite that, GSA and CRA have clear distinct goals. GSA is typically interested in the variation of a performance index with the respect to the model parameters. Since many times this is basically implemented through the derivatives, GSA tools are useful when an optimization of the system is required [9]. In the CRA algorithm, on the other hand, the purpose is not to maximize or minimize a certain optimization function, but the main interest is in finding the conditioning set, which is the subset of parameters that more likely is able to impose a specific behavior to one output of the model.

In order to facilitate the study of cancer robustness and the application of CRA, we develop the CRA Toolbox, a software package for MATLAB. It is an open source tool which allows non expert users to apply the CRA to any ODE model in a simple and quick way. The CRA Toolbox consists in a easy to use Graphical User Interface (GUI) and in a set of functions which can be easily extended by the users in order to achieve specific requirements.

In the following subsection, we briefly describe the CRA method implemented in the toolbox. A detailed description of the method can be found in [5].

**Conditional Robustness Algorithm (CRA)**

The main underlying theoretical principle of CRA is the definition of conditional robustness proposed by Kitano in [10]: it is the quantitative measure of the ability of a system S to maintain a specific property  $\tau$  against some perturbations of the parameter vector  $p$  of S. The mathematical formulation is the following:

$$R_{\tau, P}^n := \int_P f_P(p) \zeta_{\tau}^S(p) dp$$

where  $f_P(p)$  is the probability density function of  $p$ ,  $P$  is the parameter space and  $\zeta_{\tau}^S(p)$  is a function that quantifies and represents the property  $\tau$  that is under investigation.

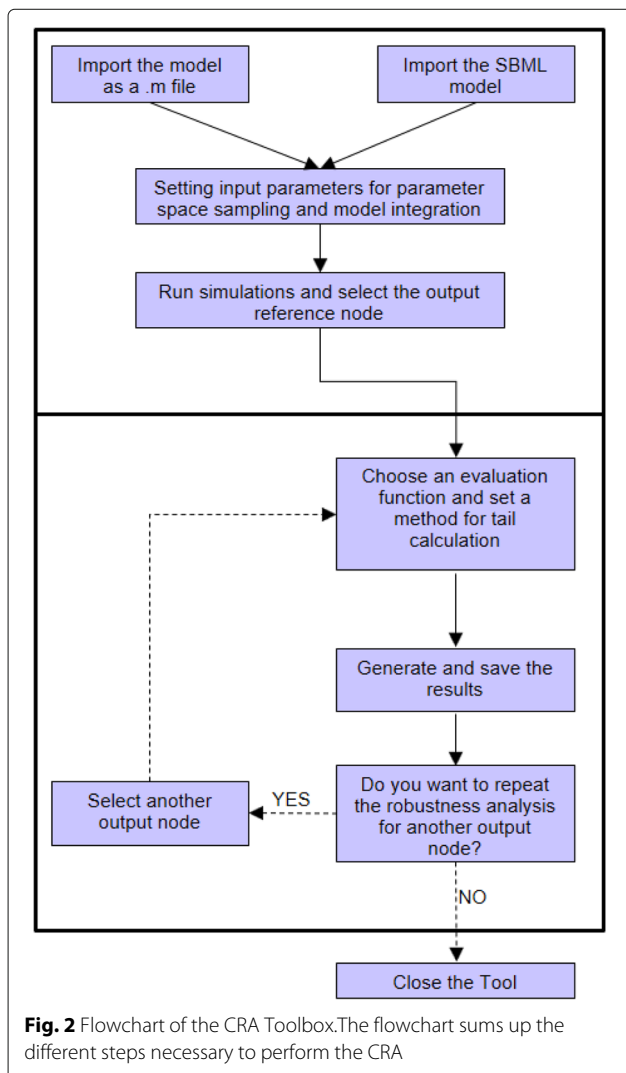
The CRA is a stochastic approach for performing conditional robustness analysis of mathematical models, such as ODE models representing biochemical interaction networks. Its purpose is to quantify the influence of each model parameter on the behavior of a specific output node. Let denote with S the following ODE system:

$$S = \begin{cases} \dot{x} = f(x, u, p), & x(0) = x_0 \\ y = h(x, p) \end{cases}$$

where  $x \in \mathbf{R}^n$  is the state space vector that contains the species included in the biological model under study;  $p \in \mathbf{R}^q$  denotes the parameter vector;  $u \in \mathbf{R}^j$  and  $y \in \mathbf{R}^m$  are the input and output vectors respectively. The key features of this algorithm are:

- simultaneous perturbation of model parameters;
- definition and computation of the evaluation function;
- estimation of the conditional probability density functions (pdfs) for each model parameter.

The parameter vector  $p$  is perturbed through Latin Hypercube sampling (LHS) and the model S is integrated for each one of the  $N_S$  samples generated in  $\mathbf{R}^q$ . This procedure allows the collection of  $N_S$  vectors of the observables  $y$ . Then, the CRA is based on the definition and the in silico computation of an evaluation function on a specific output node i.e., on a specific observable  $y_i$ . In



more details, the evaluation function can be formalized as follows:

$$\zeta : \mathbf{P} \longrightarrow \mathbf{R}, \quad z_i = \zeta_{\tau}^S(p) \tag{1}$$

where the index  $i$  represents the selected output variable of the model. Thus, the evaluation function, that depends on the time behavior of  $y_i$  (which, in turn, depends on the selected parameter vector  $p$ ), can be considered as a user defined summary statistic that stands for a specific property of the chosen output node  $y_i$ . The set of computed evaluation functions, for a specific output node  $y_i$ , has cardinality equal to  $N_S$ , i.e. the number of sampled parameter vectors. Let denote with  $f_{Z_i}(z_i)$  the pdf of the set of evaluation functions previously defined.

The CRA algorithm aims at quantifying the influence of each model parameter on a specific output node through  $f_{Z_i}(z_i)$ . In more detail, it is interested in the estimation of the distribution of the parameter vector  $p$  only when the lower and upper tail of  $f_{Z_i}(z_i)$  are selected. To this purpose, the domain of  $f_{Z_i}(z_i)$  can be partitioned into two regions by the definition of  $L(\alpha)$  and  $U(\alpha)$  as:

$$L(\alpha) = \left\{ z_i \leq a : \int_0^a f_{Z_i}(z_i) dz_i = \alpha \right\} \tag{2}$$

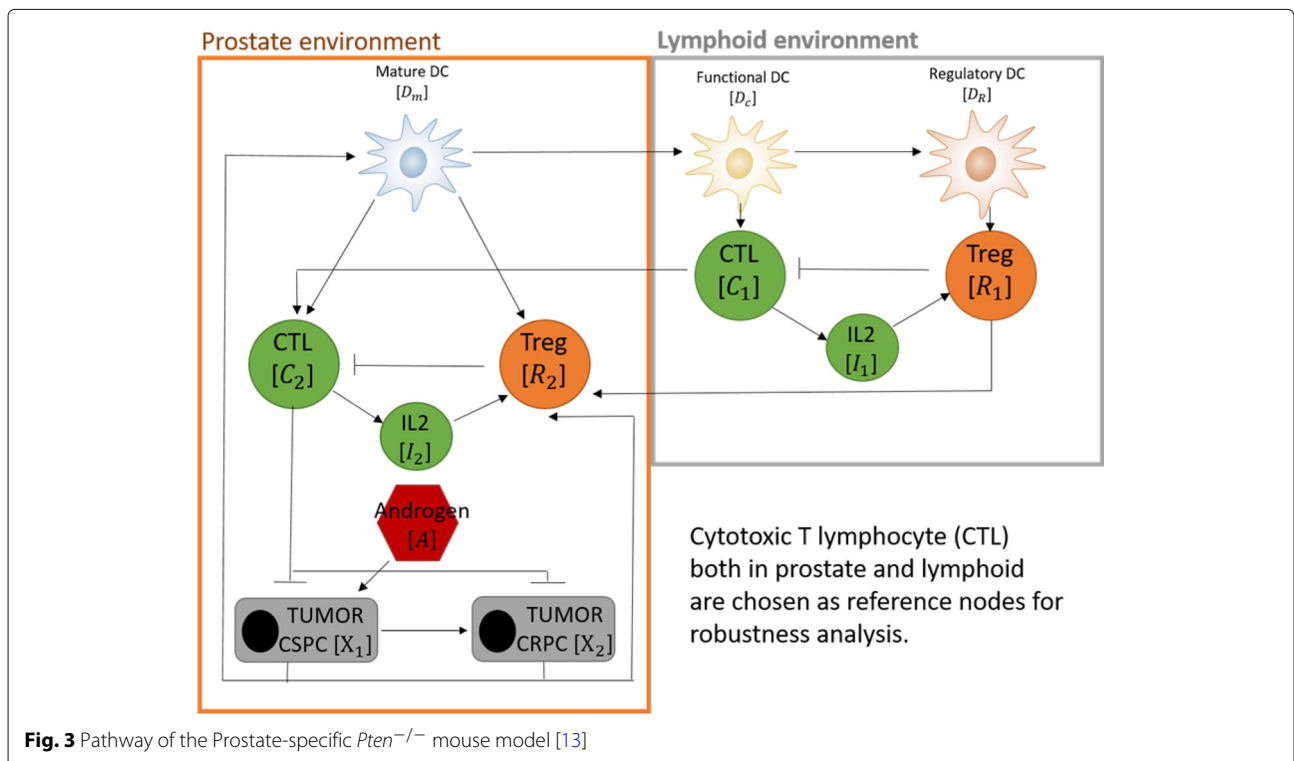
$$U(\alpha) = \left\{ z_i \geq a : \int_a^{\infty} f_{Z_i}(z_i) dz_i = \alpha \right\} \tag{3}$$

where  $\alpha$  is the level of probability that represents the area under the lower and upper tail of  $f_{Z_i}(z_i)$  and  $a$  is the corresponding threshold value in the domain of the pdf.

The partition defined in the domain of  $f_{Z_i}(z_i)$  allows the estimation of two conditional pdfs for each parameter,  $f_{p_i|L}$  and  $f_{p_i|U}$ , respectively. These two pdfs are the distributions of the parameters of  $S$  when the values of the evaluation function belong to the lower and upper tail of  $f_{Z_i}(z_i)$  respectively. Here the purpose of the estimation of  $f_{p_i|L}$  and  $f_{p_i|U}$  is to select a subset of the  $N_S$  samples of the parameters that give rise to the most divergent behaviors of the evaluation function. The two conditional densities defined above are employed for the calculus of the Moment Independent Robustness Indicator (MIRI) according to the following formula:

$$\mu_i = \int |f_{p_i|U} - f_{p_i|L}| dp_i, \quad i = 1, \dots, q \tag{4}$$

The MIRI is an index that measures the level of separation between  $f_{p_i|U}$  and  $f_{p_i|L}$  for each parameter of the model. An high value of the MIRI for a parameter  $p_i$  means that the perturbation of the parameter space along the  $p_i$  direction leads to high variation of the evaluation function. Thus, the higher the value of a MIRI, the higher is the influence of that parameter on the dynamical behavior of the selected output node.



Finally, the output of CRA is the vector  $\mu$  that contains the value of the MIRI associated to each parameter of the model. For further details about CRA, see [5].

### Implementation

The CRA Toolbox is an open source software developed as a MATLAB package. Its core is a set of functions that the user can run locally in a MATLAB environment by downloading the folder containing the toolbox. This software automates all the functionalities required by CRA to perform robustness analysis of an ODE model.

The CRA Toolbox includes a GUI that runs within MATLAB to encourage the employment of the software also for non expert users. In more detail, the tool firstly performs the import of a mathematical model written in Systems Biology Markup Language (SBML) and saved in *.xml* file format. Then, it allows the user to set the tuning parameters to regulate the parameter space perturbation and the model integration, such as the specific ODE solver to use. Before selecting the reference node from a scrollbar that lists all the outputs of the model, it is necessary to

start the simulations by clicking on a specific button. Once the in silico measures are completed, the tool requires the selection of a specific evaluation function in a predefined list and the method for the computation of the lower and higher tail of the pdf of the evaluation function. Finally, it is possible to plot and save in a user defined directory all the in silico measures, the estimated pdfs and the boxplot of MIRIs. In order to guarantee the reliability of results, the toolbox supports the generation of multiple realizations of the entire procedure and of the resulting MIRIs and pdfs. In order to speed up the model simulation we use parallel processing through the Parallel Computing Toolbox<sup>TM</sup> [11].

Moreover, we also provide an alternative implementation of the CRA Toolbox that allows the user to run the algorithm in batch mode directly from the command line. The core functions and the architecture of the software remains unchanged, but for this version we removed the GUI and we also avoided the use of Simbiology to enhance the portability of the code. Indeed, in this version of the software, the mathematical model can be given in input as a Matlab function where all the ODEs are specified and it is not required to use the SBML language and the corresponding Simbiology Object.

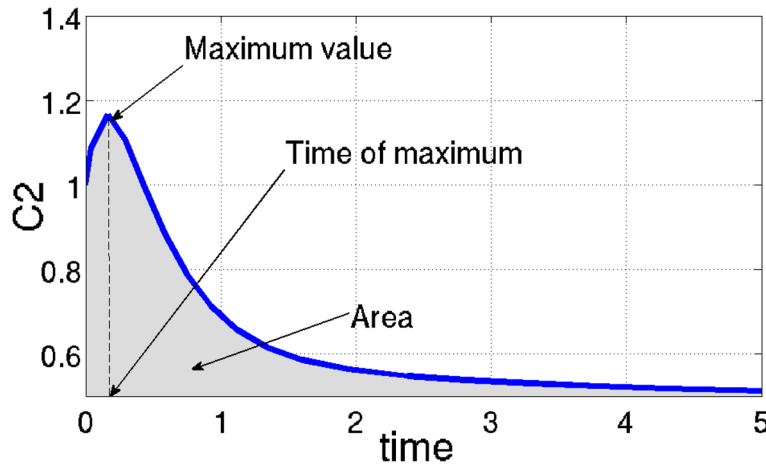
The source code of the CRA Toolbox is written according to the Object-Oriented programming paradigm as it is shown in the UML class diagram in Fig. 1. For a detailed description of all the components of the tool see (Additional file 1). The architecture of the software is modular because we implement it using software engineering design patterns to model relationships and interactions between classes. As an example, we use the behavioral Strategy pattern [12] twice between the three main

**Table 1** List of the kinetic parameters of the Prostate-specific *Pten*<sup>-/-</sup> mouse model and their corresponding nominal values [13]

Parameter name	Value
$\lambda_A$	0.6931
$r_{p1}$	0.323432
$r_{a1}$	0.1
$r_m$	6.702665
$r_{p2}$	1.426189
$r_{a2}$	1.061691
$\alpha_{XD}$	2.096368
$\alpha_{DC}$	3.077551
$\alpha_{DR}$	0.739121
$\alpha_{XR}$	0.144144
$\alpha_{CI}$	0.408862
$\alpha_{IR}$	0.354526
$\alpha_{D_C D_R}$	0.1
$\alpha_{D_R R}$	0.4722
$k_{CX}$	0.1
$k_{RC}$	1.459617
$\pi_D$	5.051123
$\pi_C$	4.111938
$\pi_R$	0.1
$\mu_C$	0.978832
$\mu_R$	0.564691
$\mu_I$	0.213819
$\mu_D$	0.650308

**Table 2** List of the initial conditions of the Prostate-specific *Pten*<sup>-/-</sup> mouse model state variables and their corresponding values [13]

State variable	Species	Value
A	Androgen	1
$X_1$	CSPC	1
$X_2$	CRPC	0
$D_m$	Mature DC	1
$C_2$	CTL in prostate	1
$R_2$	Treg in prostate	1
$I_2$	IL2 in prostate	1
$D_C$	Functional DC	1
$D_R$	Regulatory DC	1
$C_1$	CTL in lymphoid	1
$R_1$	Treg in lymphoid	1
$I_1$	IL2 in prostate	1



**Fig. 4** Evaluation functions available in the CRA Toolbox. Nominal time behavior of  $C_2$  (blue line) and the three evaluation functions measured for robustness analysis

classes of the tool: once between the *TimeBehavior* and *EvaluationFunction* classes and the other between *TimeBehavior* and *Tail* classes. This makes the code easily to extend because if a user wants to add another type of evaluation function or wants to implement another method for the pdf tail calculus, he does not have to change any other part of the code.

**Results**

In this section we show how to use the CRA Toolbox and we report the results of the application of CRA to three different ODE models: the Prostate-specific *Pten*<sup>-/-</sup> mouse model, the Pulse Generator Network and the EGFR-IGF1R pathway. The second and the third examples are used in order to verify that the CRA Toolbox produces results in agreement with those in [5]. Figure 2, Additional files 2 and 3 contain a flowchart and two images to guide the user in the use of the tool.

**Prostate-specific *Pten*<sup>-/-</sup> mouse model**

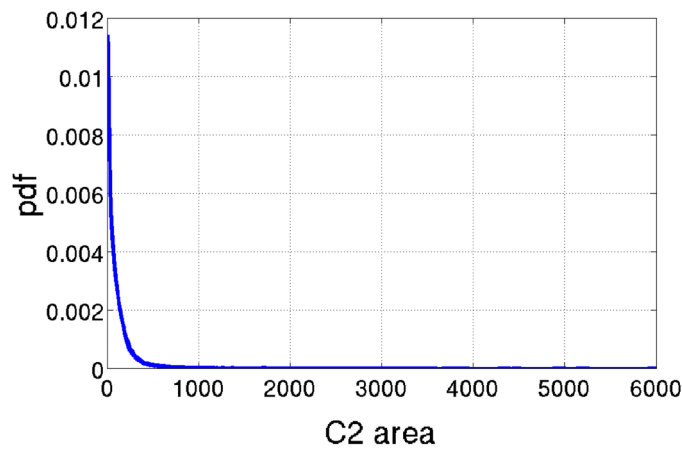
In this section, we use the ODE model proposed in [13] to illustrate the functionalities of the CRA Toolbox. This model was developed to study the interactions between prostate cancer and immune microenvironment. In more detail, it is a prostate-specific *Pten*<sup>-/-</sup> mouse model for analyzing the effect of the combined therapies with vaccines and Androgen deprivation therapy (ADT) in prostate tumor. The original model consists of two compartments, prostate and lymphoid, 11 state variables and four types of therapeutic strategies, resulting in a system of 15 ODEs and 29 parameters. The pathway of the model is shown in Fig. 3. In this example, we analyze a simplified version of this model because we consider

only the off-treatment condition (sham-castration). As a result, the mathematical model consists of 12 ODEs and 23 parameters. Equations of the model are:

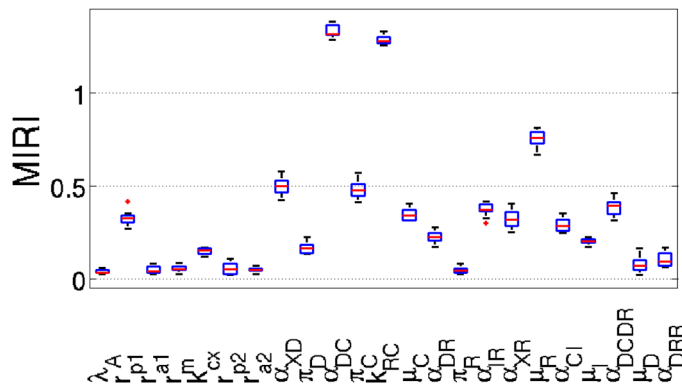
$$\begin{cases}
 \dot{A} = \lambda_A(1 - A) \\
 \dot{X}_1 = r_{p1}AX_1 - r_{a1}(1 - A)X_1 - r_m(1 - A)X_1 - k_{CX}C_2X_1 \\
 \dot{X}_2 = r_{p2}X_2 - r_{a2}X_2 + r_m(1 - A)X_1 - k_{CX}C_2X_2 \\
 \dot{D}_m = -\alpha_{XD}(r_{a1}(1 - A)X_1 + k_{CX}C_2X_1 + r_{a2}X_2 + k_{CX}C_2X_2) - \pi_D D_m \\
 \dot{C}_2 = \alpha_{DC}D_m + p_C\pi_C C_1 - k_{RC}R_2C_2 - \mu_C C_2 \\
 \dot{R}_2 = \alpha_{DR}D_m + p_R\pi_R R_1 + \alpha_{IR}I_2 + \alpha_{XR}(X_1 + X_2) - \mu_R R_2 \\
 \dot{I}_2 = \alpha_{CI}C_2 - \mu_I I_2 \\
 \dot{D}_C = p_D\pi_D D_m - \alpha_{DC}D_R D_C \\
 \dot{D}_R = \alpha_{DC}D_C - \mu_D D_R \\
 \dot{C}_1 = \alpha_{DC}D_C - \mu_C C_1 - k_{RC}R_1C_1 - \pi_C C_1 \\
 \dot{R}_1 = \alpha_{DR}D_R + \alpha_{IR}I_1 - \mu_R R_1 - \pi_R R_1 \\
 \dot{I}_1 = \alpha_{CI}C_1 - \mu_I I_1
 \end{cases} \quad (5)$$

Parameter values and initial conditions of state variables are shown, respectively, in Tables 1 and 2. Since initial condition of androgen is set to 1, i.e.  $A_0 = 1$ , the concentration of androgen keeps unchanged. Moreover, parameters  $p_C$ ,  $p_R$  and  $p_D$  represent probabilities fixed all to 0.5 in [13] and thus they are not perturbed in the CRA procedure.

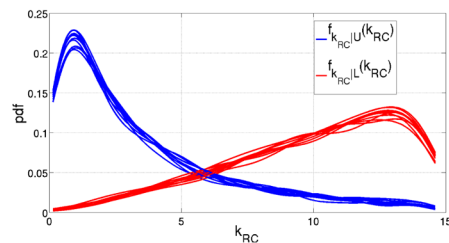
We apply the CRA Toolbox to the ODE model by setting tuning parameters of the procedure as follows: number of samples  $N_S$  equal to 10000, lower boundary and upper boundary of the LHS equal to 0.1 and 10 respectively. We run 100 independent realizations to verify the reliability and stability of the procedure. We choose different output



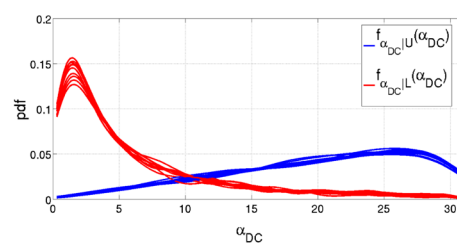
(a)



(b)



(c)



(d)

**Fig. 5** Results of the Prostate-specific *Pten*<sup>-/-</sup> mouse model:  $C_2$  area. Output of the CRA Toolbox when the area under the curve of  $C_2$  time behavior is chosen as evaluation function. **a** Pdf of the area under the curve of  $C_2$ . **b** Boxplot of the 100 realizations of the MIRIs. **c-d** Conditional pdfs of the parameters with a MIRI value above 1

nodes and evaluation functions in order to show results of the CRA Toolbox in a complete and comprehensive way. Specifically, we select as output nodes both variables  $C_2$  and  $C_1$ , which represent cytotoxic T lymphocyte (CTL) in prostate and lymphoid respectively. For  $C_2$ , we measure all the three evaluation functions offered by the software, i.e. the area under the curve, the maximum and the time of maximum reached by the time behavior of CTL, as

shown in Fig. 4. Conversely, for  $C_1$ , we perform robustness analysis using as evaluation function only the area under the curve. In all cases, we set equal to 1000 the dimension of the upper and lower tail of the evaluation function pdf, in order to guarantee a stable estimation of the conditional parameter pdfs [5].

Results of CRA applied to variable  $C_2$  to measure the area are shown in Fig. 5. Parameters  $\alpha_{DC}$  and  $k_{RC}$  have

MIRI values above 1, thus they have a great impact on the chosen evaluation function. Other parameters influencing the selected output are  $\mu_R$ ,  $\alpha_{XD}$  and  $\pi_C$ , having MIRI values between 0.5 and 1.

Similar results are obtained for the maximum value of  $C_2$ , as shown in Fig. 6. Parameters with the highest MIRI values ( $\sim 1$ ) are  $k_{RC}$  and  $\alpha_{DC}$  as before, while parameters  $r_{p1}$ ,  $\alpha_{XD}$  and  $\mu_R$  have all MIRIs around 0.5.

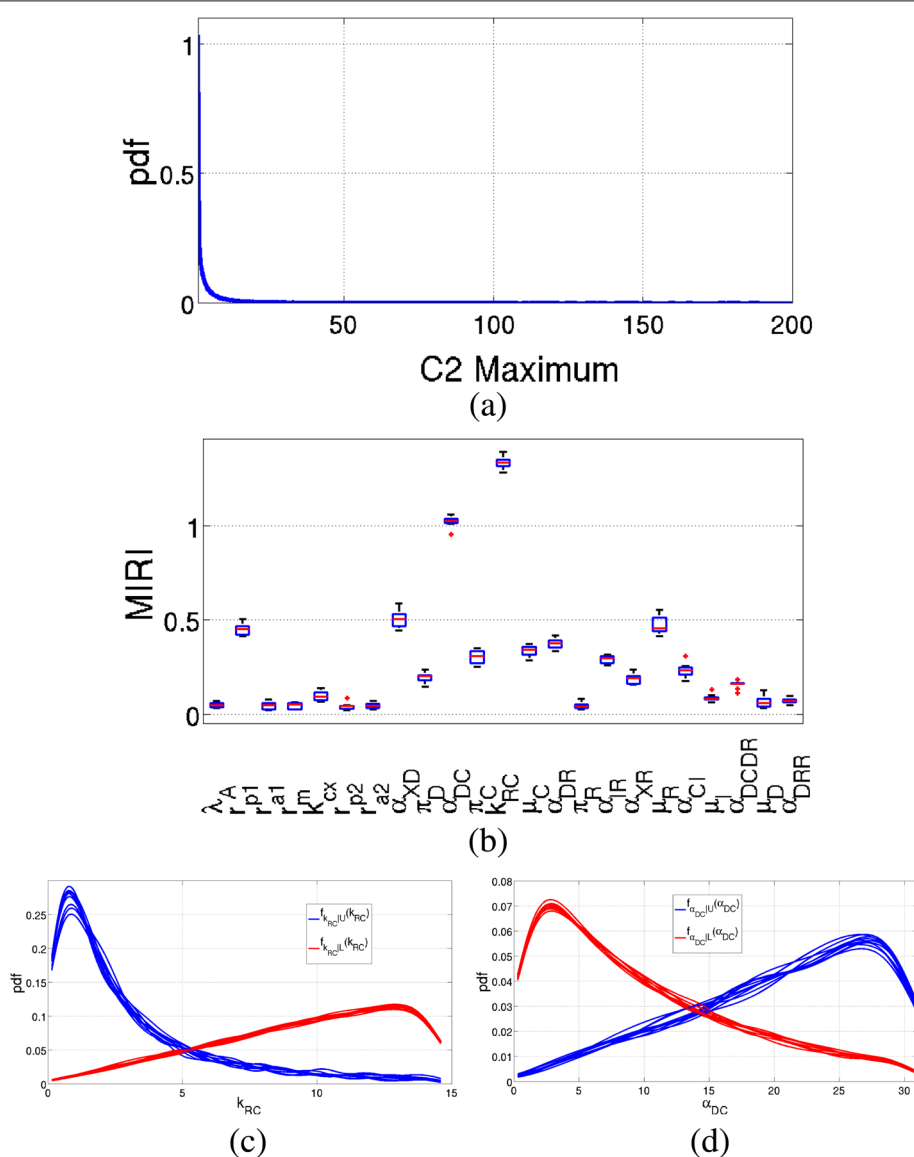
Figure 7 shows results obtained for the time of maximum of variable  $C_2$ . In this case, MIRIs have values lower if compared to the previous examples. The most influential parameters are  $\alpha_{DC}$ ,  $k_{RC}$  and  $\mu_R$ , with MIRI values around 0.6.

As regards variable  $C_1$ , results of the measured area are shown in Fig. 8. Two are the parameters with MIRIs above 1:  $r_{p1}$  and  $\alpha_{DC}$ . All the remaining parameters have low values, except for  $\alpha_{XD}$  and  $\alpha_{DCDR}$  with values around 0.5.

Table 5 contains the time necessary to perform all the simulations.

### Pulse generator network

We test the CRA Toolbox on a small toy system belonging to the field of Synthetic Biology because synthetic models are one of the best examples of the importance of theoretical modeling in the biological reality [14]. The



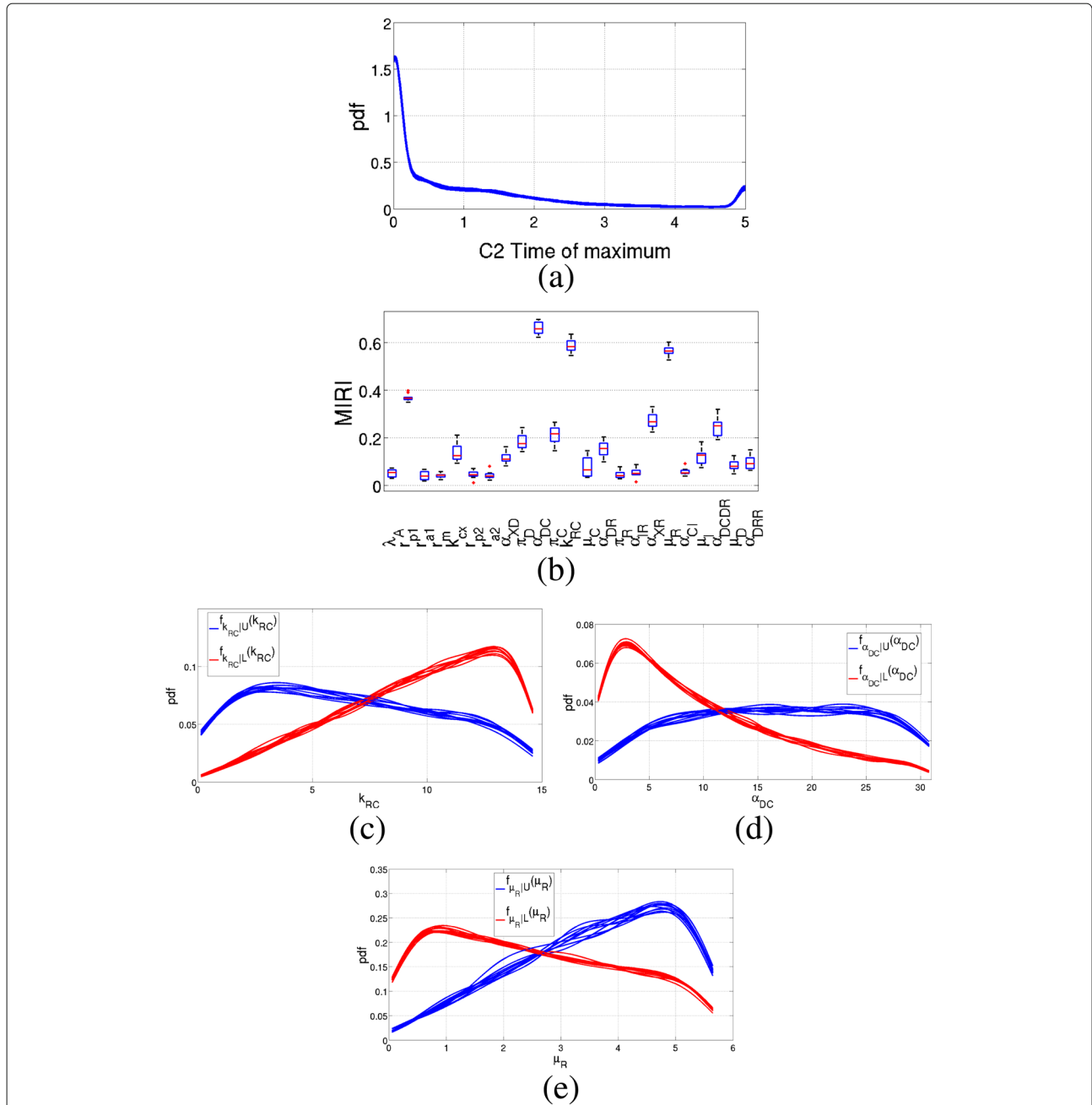
**Fig. 6** Results of the Prostate-specific  $Pten^{-/-}$  mouse model:  $C_2$  maximum value. Output of the CRA Toolbox when the maximum value of  $C_2$  time behavior is chosen as evaluation function. **a** Pdf of the maximum value of  $C_2$ . **b** Boxplot of the 100 realizations of the MIRIs. **c-d** Conditional pdfs of the parameters with a MIRI value above 1



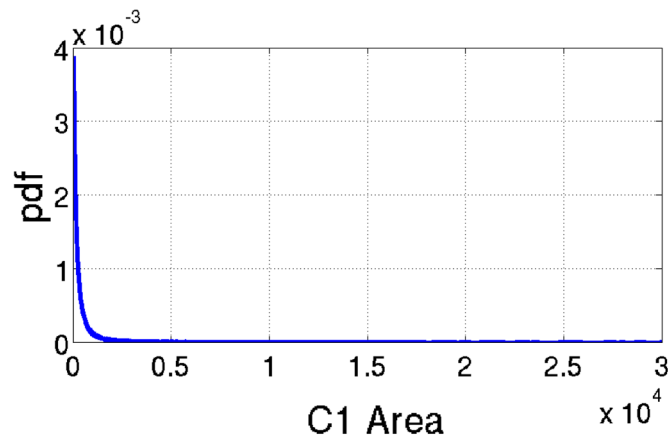
pulse generator network consists of three nodes, representing three genes, aimed at producing a transient output response to a persistent input stimulus [14]. Figure 9 shows the interaction schema of the model. Node S1 is the input step signal that activates both R2 and Y. R2 is the so called repressor because it acts as a deactivator of the product Y. The corresponding ODE model has two state

variables, eight kinetic parameters and one input signal. The following ODEs are written using a Hill function for the activation and repression function [5, 15]:

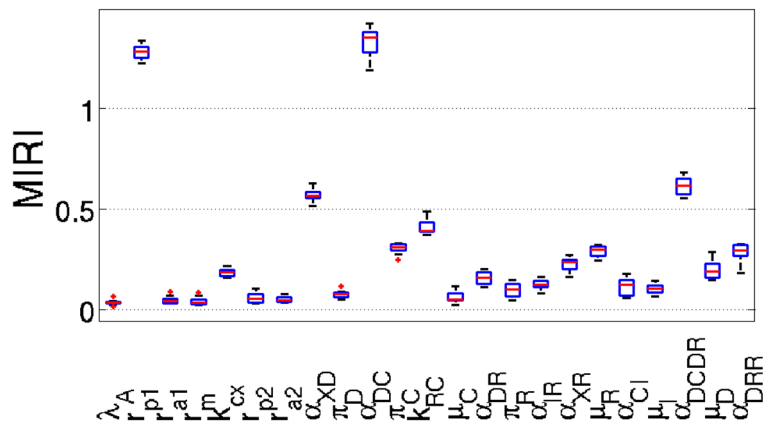
$$\begin{cases} \dot{R}_2 = k_1 \frac{(S_1/K_1)^{n_1}}{1+(S_1/K_1)^{n_1}} - \lambda_2 R_2 \\ \dot{Y} = \frac{k_{12}}{1+(R_2/K_2)^{n_2}} \frac{(S_1/K_1)^{n_1}}{1+(S_1/K_1)^{n_1}} - \lambda Y. \end{cases} \quad (6)$$



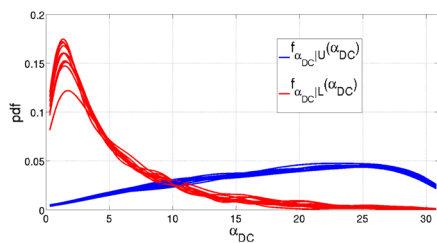
**Fig. 7** Results of the Prostate-specific *Pten*<sup>-/-</sup> mouse model: C<sub>2</sub> time of maximum. Output of the CRA Toolbox when the time of maximum of C<sub>2</sub> time behavior is chosen as evaluation function. **a** Pdf of the time of maximum of C<sub>2</sub>. **b** Boxplot of the 100 realizations of the MIRIs. **c-d-e** Conditional pdfs of the parameters with a MIRI value above 0.5



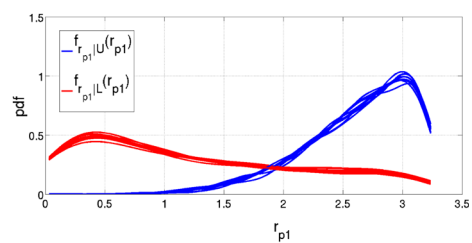
(a)



(b)



(c)

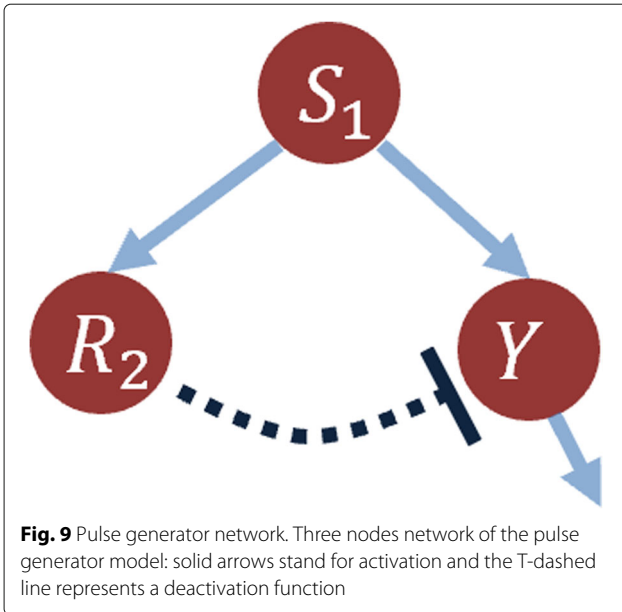


(d)

**Fig. 8** Results of the Prostate-specific *Pten*<sup>-/-</sup> mouse model: C<sub>1</sub> area. Output of the CRA Toolbox when the area under the curve of C<sub>1</sub> time behavior is chosen as evaluation function. **a** Pdf of the area of C<sub>1</sub>. **b** Boxplot of the 100 realizations of the MIRIs. **c-d** Conditional pdfs of the parameters with a MIRI value above 1

Nominal values for the parameters in Eq. 6 are  $k_1=5$  nM/min,  $k_{12}=20$  nM/min,  $\lambda_2=0.01$  nM/min,  $\lambda=0.04$  nM/min,  $K_1=1$  nM,  $K_2=100$  nM and  $n_1=n_2=3$ . We run the CRA Toolbox setting the tuning parameters as reported in [5]. More in detail, we set the number of realizations, the lower boundary, the upper boundary and the number of samples equal to 100, 0.1, 10 and 10000 respectively.

Parameters  $n_1$  and  $n_2$  remain fixed to their nominal values since they are not included in robustness analysis. We select as reference node the observable Y and we progressively perform the CRA using all the three evaluation functions provided by the tool, i.e. the area under the curve, the maximum value and the time of maximum. Figure 10 shows the pdf of the area and the corresponding



boxplot of MIRIs when choosing 1000 as the dimension of both the lower and higher tails. The other subfigures of Fig. 10 show the conditional pdfs of each parameter that are used for the calculus of MIRIs. Figure 11 shows the results of robustness analysis when the maximum of output variable Y is selected as evaluation function. Figure 12 reports the output generated by the CRA Toolbox about the time of maximum of variable Y. From Figs. 10b, 11b and 12b, it is clear how parameter  $\lambda$  has a great impact on the output behavior of Y, since it influences all the three evaluation functions with a high value of the corresponding MIRI. Moreover, parameter  $k_{12}$  is the one with highest value of the MIRI in Fig. 11b and thus it especially influences the maximum value reached by the Y node. Finally, parameter  $K_2$  is the most relevant when dealing with the time of maximum of variable Y because it has the highest MIRI value in Fig. 12b. Table 5 reports the time required to complete all the simulations described in this section.

### EGFR-IGF1R pathway in lung cancer

The last example of application of the CRA Toolbox is the EGFR-IGF1R pathway, which is one of the most relevant interaction network for the study of cancer pathogenesis and progression in Non-Small Cell Lung Cancer (NSCLC). Figure 13 depicts the pathway. For detailed information about the biological importance of this network and the role of the different nodes see [16]. The corresponding mathematical ODE model is published in [17] (id:MODEL1209230000). The dynamical ODE model is composed of ten equations (Eq. 7), containing two types of kinetic laws: the law of mass action and the Michaelis-Menten kinetics.

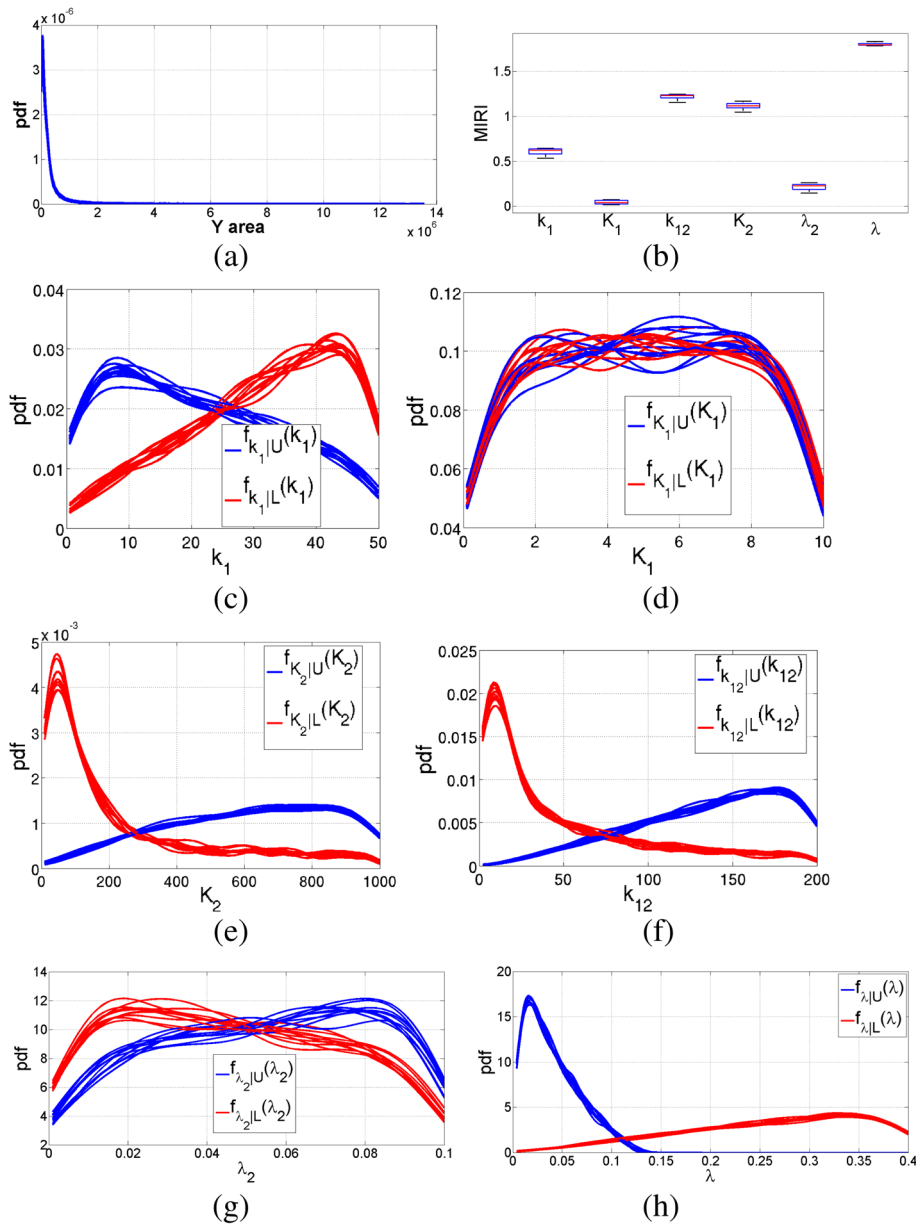
$$\begin{cases} \dot{x}_1 = -p_1x_1 \\ \dot{x}_2 = -p_2x_2 \\ \dot{x}_3 = p_6x_1 \frac{x_3^T - x_3}{p_7 + x_3^T - x_3} + p_{14}x_2 \frac{x_3^T - x_3}{p_{15} + x_3^T - x_3} - p_{12}x_8 \frac{x_3}{p_{13} + x_3} \\ \dot{x}_4 = p_8x_3 \frac{x_4^T - x_4}{p_9 + x_4^T - x_4} - p_{33}u_3 \frac{x_4}{p_{34} + x_4} \\ \dot{x}_5 = p_{27}x_4 \frac{x_5^T - x_5}{p_{28} + x_5^T - x_5} - p_{37}u_1 \frac{x_5}{p_{38} + x_5} - p_{31}x_{10} \frac{x_5}{p_{32} + x_5} \\ \dot{x}_6 = p_{29}x_5 \frac{x_6^T - x_6}{p_{30} + x_6^T - x_6} - p_{35}u_2 \frac{x_6}{p_{36} + x_6} \\ \dot{x}_7 = p_{10}x_6 \frac{x_7^T - x_7}{p_{11} + x_7^T - x_7} - p_{23}u_2 \frac{x_7}{p_{24} + x_7} \\ \dot{x}_8 = p_4x_7 \frac{x_8^T - x_8}{p_5 + x_8^T - x_8} - p_{39}x_8 \\ \dot{x}_9 = p_{25}x_4 \frac{x_9^T - x_9}{p_{26} + x_9^T - x_9} + p_{16}x_2 \frac{x_9^T - x_9}{p_{17} + x_9^T - x_9} + p_{18}x_1 \frac{x_9^T - x_9}{p_{19} + x_9^T - x_9} - p_3x_9 \\ \dot{x}_{10} = p_{20}x_9 \frac{x_{10}^T - x_{10}}{p_{21} + x_{10}^T - x_{10}} - p_{22}x_{10} \end{cases} \quad (7)$$

Table 3 explicits names and initial concentrations of the ten state variables, of the three input signals and of the eight total protein parameters. Table 4 lists all the kinetic model parameters and their nominal values. We run the CRA Toolbox according to the guidelines in [5]: the number of realizations, the lower boundary, the upper boundary and the number of samples are set equal to 100, 0.1, 10 and 10000 respectively. We select as reference node the active form of ERK protein, since it is one of the best indicator of the proliferation attitude of lung cancer. More in detail, we perform robustness analysis of the model choosing as evaluation function the area under the curve of the time behavior of ERK. Figure 14a shows the pdf of the evaluation function and Fig. 14b the boxplot for each one of the 39 MIRI parameters. This figure clearly shows that four parameters have a MIRI higher than 0.6 and thus they significantly influence the dynamical behavior of the output node. More in detail, these parameters are:  $k_{ERK-PP2A}$ ,  $K_{MERK-PP2A}$ ,  $k_{MEK-PP2A}$  and  $K_{MMEK-PP2A}$  and Fig. 14c, d, e and f show the corresponding conditional pdfs.

Table 5 reports the time required to complete the robustness analysis through the CRA algorithm.

### Discussion

The CRA is an algorithm to study the robustness of complex biological networks and it allows the identification of few parameters that have a major impact on the behavior of a selected output variable. One of its main innovations is the introduction of a sensitivity measure, the MIRI, that takes advantage of all the conditional distributions of the parameters, without reference to a specific moment. In [18] there are the mathematical details of this class of indicators and the comparison with the variance-based uncertainty importance measures.

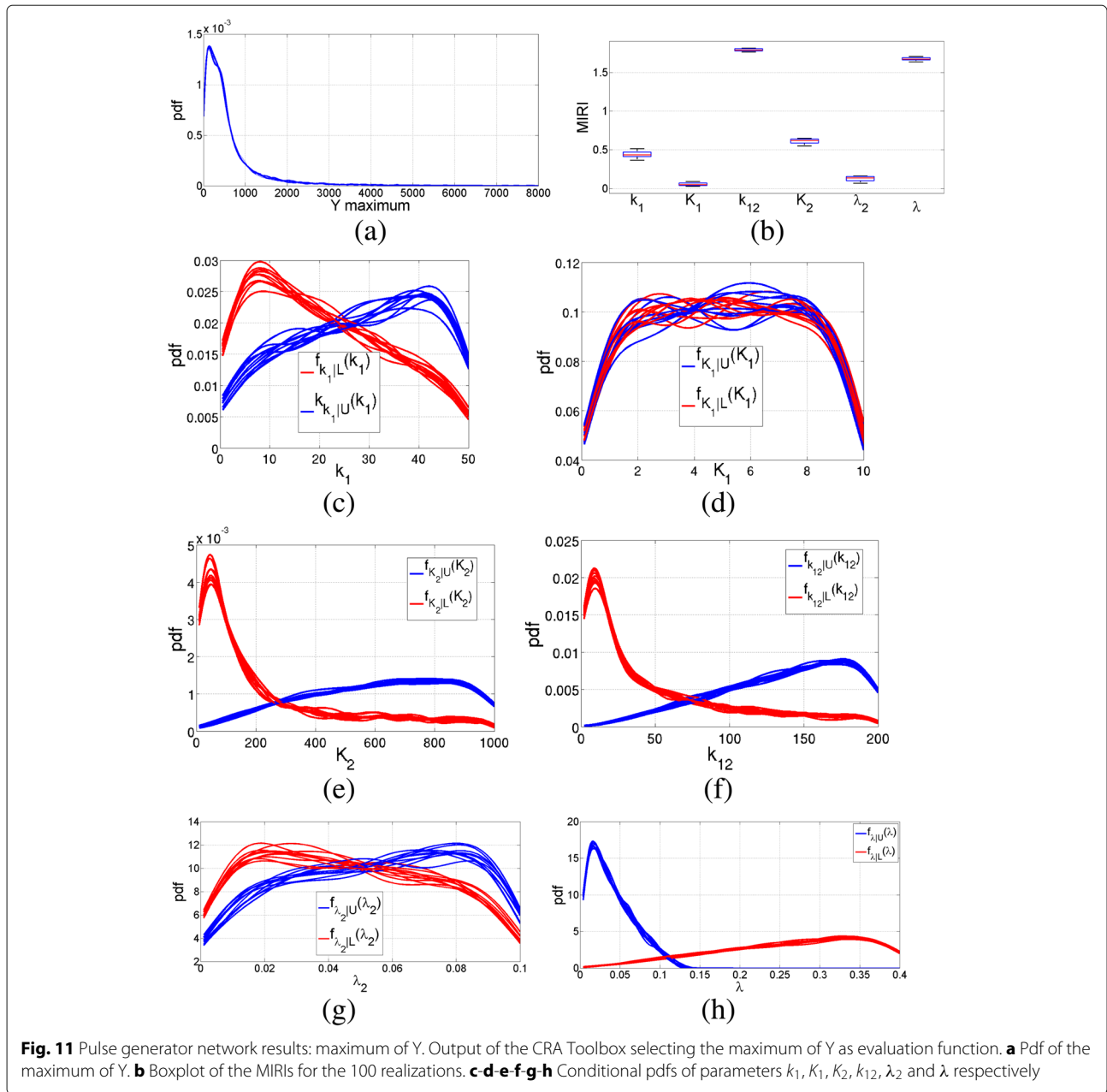


**Fig. 10** Pulse generator network results: Y area. Output of the CRA Toolbox selecting the area of Y as evaluation function. **a** Pdf of the area of Y. **b** Boxplot of the MIRIs for the 100 realizations. **c-d-e-f-g-h** Conditional pdfs of parameters  $k_1$ ,  $K_1$ ,  $K_2$ ,  $k_{12}$ ,  $\lambda_2$  and  $\lambda$  respectively

Here we present the CRA Toolbox, a software package for MATLAB aimed at performing the robustness analysis based on the paradigm of the CRA. The tool consists of a set of MATLAB functions that automate all the necessary mathematical steps from the integration of an ODE model until the calculus of the MIRIs. We decide to use the Object-Oriented programming paradigm because it allows us the development of an extendable and flexible architecture through the implementation of different

engineering design patterns. In this way, other users can add blocks of software to define novel evaluation functions and other methods for the identification of the pdf tails without modifying the structure of the software and the source code of the existing classes.

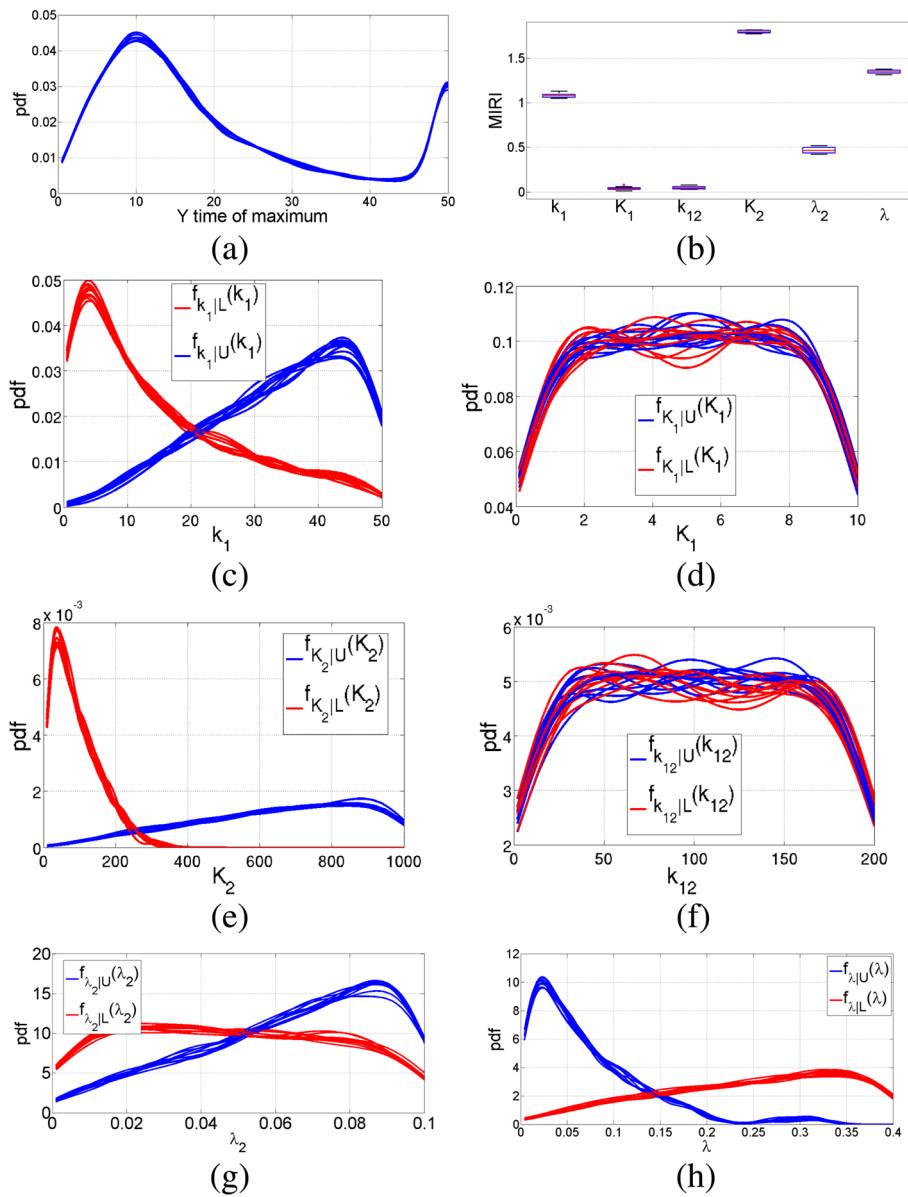
Here we tested the CRA on three ODE models that contain different kinetic laws and different number of state variables. All the simulations were performed on a



Intel Core i7-4700HQ CPU, 2.40GHz 8, 16-GB memory, Ubuntu 16.04 LTS (64bit). From Table 5 it is clear how the time to complete a realization of the CRA highly depends both on the dimension of the model and on the number of samples  $N_S$  of the LHS. Indeed, from a computational point of view, the most intensive part of the algorithm is the integration of the mathematical model  $N_S$  times. Moreover, the choice of  $N_S$  influences the estimation of the evaluation function pdf, whose cost increases with  $N_S$ . On the other hand, the computation of the MIRIs is pretty fast, independently of the type of evaluation

function selected. For more details about the choice of the tuning parameters and the computational costs of the CRA for different settings of tuning parameters see Additional file 4.

One of the key ideas in [5] is that the theoretical statements of the CRA do not depend on the specific modeling technique used to represent a biological phenomenon. For this reason, one of the possible future development of the tool is to augment the number and types of formats of mathematical models taken in input. Finally, the code of the CRA Toolbox can be easily adapted to other



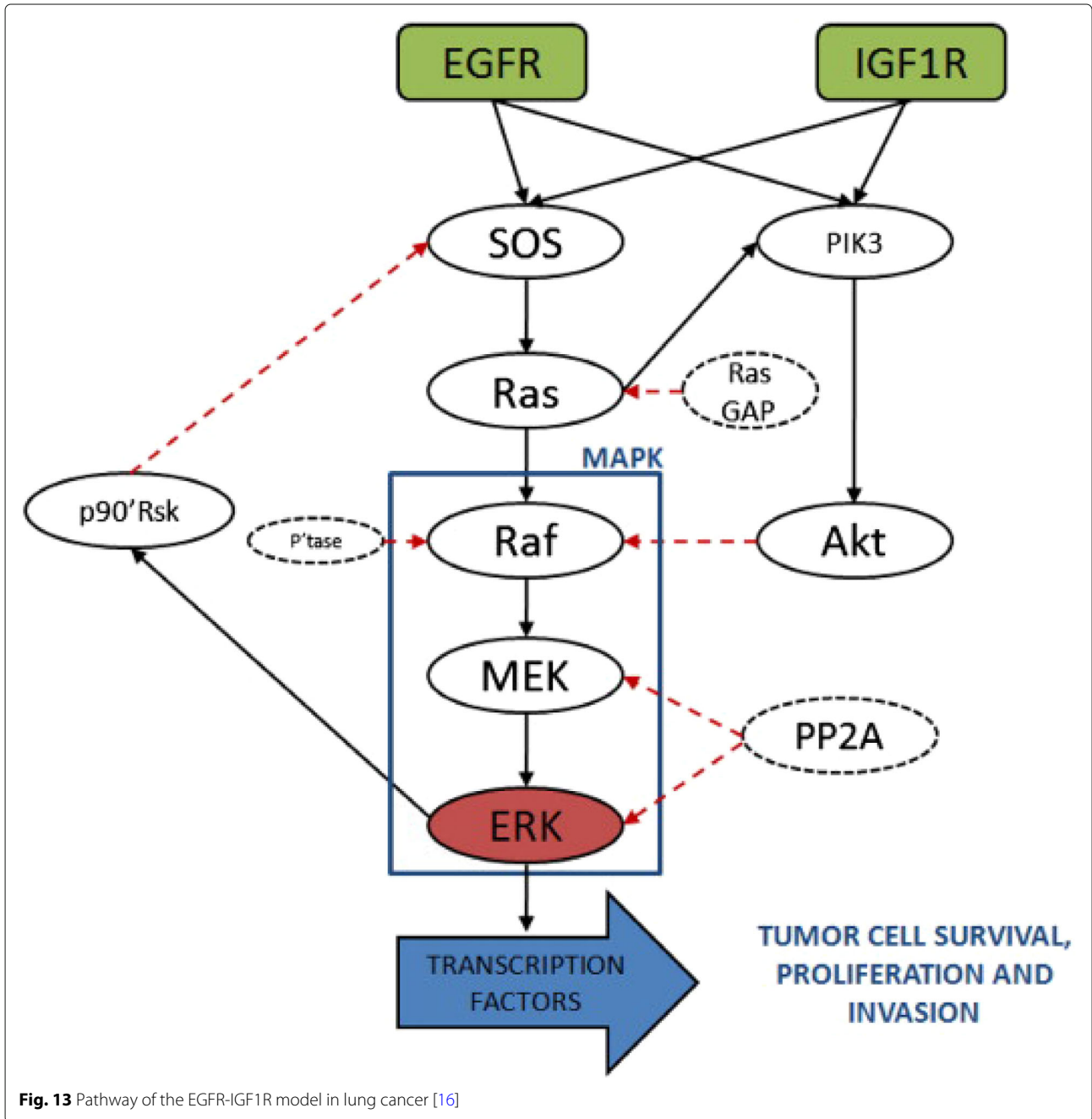
**Fig. 12** Pulse generator network results: time of maximum of Y. Output of the CRA Toolbox selecting the time of maximum of Y as evaluation function. **a** Pdf of the time of maximum of Y. **b** Boxplot of the MIRIs for the 100 realizations. **c-d-e-f-g-h** Conditional pdfs of parameters  $k_1$ ,  $K_1$ ,  $K_2$ ,  $k_{12}$ ,  $\lambda_2$  and  $\lambda$  respectively

open source programming languages such as Octave and Python.

### Conclusions

The CRA Toolbox is unique in the category of the robustness analysis tools because it is the specific implementation of the CRA with all its features. It has the scope to enlarge and facilitate the usage of this algorithm and to disclose it also to non expert users. This can

significantly help the oncological research of physicians in discover novel targeted therapies. Moreover, in [19], the Conditional Robustness Calibration (CRC) algorithm is presented, which is an upgrade of the CRA that allows the generation of MIRIs for a given model taking into account multiple nodes simultaneously. This novel algorithm suggests that the CRA Toolbox can be modified and extended in the future in order to implement also the functionalities of CRC.



**Fig. 13** Pathway of the EGFR-IGF1R model in lung cancer [16]

**Table 3** List of species included in the EGFR-IGF1R model and the corresponding initial concentrations and total protein amount

State Variable	Species	Value
$x_1$	<i>EGFR*</i>	8000
$x_2$	<i>IGF1R*</i>	650
$x_3$	<i>SOS</i>	0
$x_4$	<i>Ras*</i>	0
$x_5$	<i>Raf*</i>	0
$x_6$	<i>MEK*</i>	0
$x_7$	<i>Erk*</i>	0
$x_8$	<i>p90Rsk*</i>	0
$x_9$	<i>PIK3*</i>	0
$x_{10}$	<i>Akt*</i>	0
$u_1$	<i>RafPP</i>	120000
$u_2$	<i>PP2A</i>	120000
$u_3$	<i>RasGab</i>	120000
$x_3^T$	<i>DSOS</i>	120000
$x_4^T$	<i>Ras</i>	120000
$x_5^T$	<i>Raf</i>	120000
$x_6^T$	<i>MEK</i>	600000
$x_7^T$	<i>Erk</i>	600000
$x_8^T$	<i>p90Rsk</i>	120000
$x_9^T$	<i>PIK3</i>	120000
$x_{10}^T$	<i>Akt</i>	120000

**Table 4** List of the kinetic parameters of the EGFR-IGF1R model and their corresponding nominal values

Parameter	Name	Value
$p_1$	$\gamma_{EGFR}$	0.02
$p_2$	$\gamma_{IGF1R}$	0.02
$p_3$	$kd_{PIK3}$	0.005
$p_4$	$k_{p90Rsk-Erk}$	0.0213697
$p_5$	$KM_{p90Rsk-Erk}$	763523
$p_6$	$k_{SOS-EGFR}$	694.731
$p_7$	$KM_{SOS-EGFR}$	6086070
$p_8$	$k_{Ras-SOS}$	32.344
$p_9$	$KM_{Ras-SOS}$	35954.3
$p_{10}$	$k_{Erk-MEK}$	9.85367
$p_{11}$	$KM_{Erk-MEK}$	1007340
$p_{12}$	$k_{DSOS-p90Rsk}$	161197
$p_{13}$	$KM_{DSOS-p90Rsk}$	896896
$p_{14}$	$k_{SOS-IGFR}$	500
$p_{15}$	$KM_{SOS-IGFR}$	1000000
$p_{16}$	$k_{PIK3-IGF1R}$	10.6737
$p_{17}$	$KM_{PIK3-IGF1R}$	184912
$p_{18}$	$k_{PIK3-EGFR}$	10.6737
$p_{19}$	$KM_{PIK3-EGFR}$	184912
$p_{20}$	$k_{Akt-PIK3}$	0.0566279
$p_{21}$	$KM_{Akt-PIK3}$	653951
$p_{22}$	$kd_{Akt}$	0.005
$p_{23}$	$k_{Erk-PP2A}$	9.85367
$p_{24}$	$KM_{Erk-PP2A}$	1007340
$p_{25}$	$k_{PIK3-Ras}$	0.0771067
$p_{26}$	$KM_{PIK3-Ras}$	272056
$p_{27}$	$k_{Raf-Ras}$	0.884096
$p_{28}$	$KM_{Raf-Ras}$	62464.6
$p_{29}$	$k_{Raf-MEK}$	185.759
$p_{30}$	$KM_{Raf-MEK}$	4768350
$p_{31}$	$k_{Raf-Akt}$	15.1212
$p_{32}$	$KM_{Raf-Akt}$	119355
$p_{33}$	$k_{Ras-RasGab}$	1509.36
$p_{34}$	$KM_{Ras-RasGab}$	1432410
$p_{35}$	$k_{MEK-PP2A}$	2.83243
$p_{36}$	$KM_{MEK-PP2A}$	518753
$p_{37}$	$k_{Raf-RafPP}$	0.126329
$p_{38}$	$KM_{Raf-RafPP}$	1061.71
$p_{39}$	$kd_{p90Rsk}$	0.005





**Table 5** Computational cost to run the CRA algorithm in all the examples provided in the Results section

Model	Output node	Evaluation Function	Time (sec.)
Pulse Generator Network	Y	Area	45
		Maximum	45
		Time of Maximum	42
Prostate-specific <i>Pten</i> <sup>-/-</sup> mouse model	C2	Area	1702
		Maximum	1703
		Time of Maximum	1706
EGFR-IGF1R pathway in lung cancer	ERK	Area	1708
		Area	102

The times are expressed in seconds and they refer to a single realization of the CRA

## Availability data and requirements

**Project name:** CRA Toolbox

**Project home page:** <http://gitlab.ict4life.com/SysBioThe/CRA-Matlab>

**Operating system(s):** Platform independent

**Programming language:** MATLAB

**Other requirements:** MATLAB

**License:** MATLAB

**Any restrictions to use by non-academics:** License needed for MATLAB

## Additional files

**Additional file 1:** This *.pdf* file is a detailed description of all the classes and methods implemented in the tool and it is a useful guide for users to understand how to run and use the software package. (PDF 62 kb)

**Additional file 2:** This *.png* file is a screenshot of the first part of the GUI of the CRA Toolbox. (PNG 47 kb)

**Additional file 3:** This *.png* file is a screenshot of the second part of the GUI of the CRA Toolbox. (PNG 60 kb)

**Additional file 4:** This *.pdf* file is the user guide of the CRA Toolbox. (PDF 327 kb)

## Acknowledgements

The authors would like to thank ICT4life srl group for the helpful discussions about implementation, potential extensions/uses of modules within the algorithm and make freely available a public project with GitLab ICT4life repository.

We thank Italian Association of Cancer Research (AIRC) for supporting this work.

## Authors' contributions

FB, CA, and LT conceived and designed the toolbox. FB, CA and LT wrote the code. CA and LT wrote the user's documentation and developed the examples. FB, CA, LT and PV wrote the paper. All authors read and approved the final manuscript.

## Funding

CA and TL are funded by Italian Association for Cancer Research (AIRC) grant 15713/2014. The funding body had no role in the design of the study and collection, analysis, interpretation of data and in writing the manuscript.

## Availability of data and materials

The most recent version of the software and the documentation can be found in a public available GitLab repository using the following link: <http://gitlab.ict4life.com/SysBioThe/CRA-Matlab>.

## Ethics approval and consent to participate

Not applicable.

## Consent for publication

Not applicable

## Competing interests

The authors declare that they have no competing interests.

## Author details

<sup>1</sup>Independent Researcher, Belvedere 44, 06036 Montefalco, Perugia, Italy.

<sup>2</sup>Department of Engineering, University of Perugia, G. Duranti 95, 06132 Perugia, Italy.

Received: 26 April 2019 Accepted: 5 June 2019

Published online: 09 July 2019

## References

1. Kitano H. Systems biology: a brief overview. *Science*. 2002;295(5560):1662–4.

2. Ghosh S, Matsuoka Y, Asai Y, Hsin K-Y, Kitano H. Software for systems biology: from tools to integrated platforms. *Nat Rev Genet*. 2011;12(12):821.
3. Werner HM, Mills GB, Ram PT. Cancer systems biology: a peek into the future of patient care?. *Nat Rev Clin Oncol*. 2014;11(3):167.
4. Kitano H. Biological robustness. *Nat Rev Genet*. 2004;5(11):826.
5. Bianconi F, Baldelli E, Luovini V, Petricoin EF, Crinò L, Valigi P. Conditional robustness analysis for fragility discovery and target identification in biochemical networks and in cancer systems biology. *BMC Syst Biol*. 2015;9(1):70.
6. Dayarian A, Chaves M, Sontag ED, Sengupta AM. Shape, size, and robustness: feasible regions in the parameter space of biochemical networks. *PLoS Comput Biol*. 2009;5(1):1000256.
7. Kwon Y-K, Cho K-H. Quantitative analysis of robustness and fragility in biological networks based on feedback dynamics. *Bioinformatics*. 2008;24(7):987–94.
8. Iooss B, Lemaitre P. A review on global sensitivity analysis methods. In: Uncertainty management in simulation-optimization of complex systems; 2015. p. 101–22.
9. Rizk A, Batt G, Fages F, Soliman S. A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*. 2009;25(12):169–78.
10. Kitano H. Towards a theory of biological robustness. *Mol Syst Biol*. 2007;3(1):137.
11. Perform Parallel Computations on Multicore Computers, GPUs, and Computer Clusters. <https://it.mathworks.com/products/parallel-computing.html>.
12. Helm R, Johnson RE, Gamma E, Vlassides J. Design Patterns: Elements of Reusable Object-oriented Software: Addison Wesley; 2000.
13. Peng H, Zhao W, Tan H, Ji Z, Li J, Li K, Zhou X. Prediction of treatment efficacy for prostate cancer using a mathematical model. *Sci Rep*. 2016;6:21599.
14. Szallasi Z, Stelling J, Periwal V. System Modeling in Cellular Biology: the MIT Press; 2006.
15. Alon U. Network motifs: theory and experimental approaches. *Nat Rev Genet*. 2007;8(6):450.
16. Bianconi F, Baldelli E, Ludovini V, Crinò L, Flacco A, Valigi P. Computational model of egfr and igf1r pathways in lung cancer: a systems biology approach for translational oncology. *Biotechnol Adv*. 2012;30(1):142–53.
17. Li C, Donizelli M, Rodriguez N, Dharuri H, Endler L, Chelliah V, Li L, He E, Henry A, Stefan MI, et al. Biomedels database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst Biol*. 2010;4(1):92.
18. Borgonovo E. A new uncertainty importance measure. *Reliab Eng Syst Saf*. 2007;92(6):771–84.
19. Bianconi F, Antonini C, Tomassoni L, Valigi P. Robust calibration of high dimension nonlinear dynamical models for omics data: An application in cancer systems biology. *IEEE Trans Control Syst Technol*. 2018;99:1–12.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.