UNIVERSITY OF MILANO-BICOCCA

Department of Informatics, Systems and Communication
Ph.D. Program in Computer Science – Cycle XXXII

DOCTORAL THESIS

Fuzzy logic for the modeling and simulation of complex systems

---

Simone Spolaor

Supervisors:
Prof. Daniela Besozzi
Dr. Marco S. Nobile

Academic year 2018-2019

Abstract

Complex systems typically display emergent dynamic behaviors, which cannot be outlined by a mere description of their constituting parts. Such systems are ubiquitous in natural and artificial settings, ranging from biology and physics, to engineering and economics, and spark the interest of many scientists belonging to different research fields. To gain better insights on their inner working, complex systems are studied by means of computational methods, which allow to model and reproduce their behavior *in silico*. As a consequence, several mathematical formalisms were developed in the last decades to model complex systems, capture different features of their dynamic behavior, and leverage any available quantitative or qualitative data about their components. The aim of this thesis is to show how fuzzy logic can be exploited to overcome some of the limitations that still affect the modeling of complex systems, namely: predict emergent dynamic behaviors even when there is a lack of precise quantitative information; deal with the presence of heterogeneous systems components, spanning different levels of temporal, spatial of functional organization; bridge the gap between quantitative and qualitative modeling, in order to define hybrid models that can simultaneously exploit the peculiar advantages provided by both approaches. This thesis presents two novel modeling frameworks, based on fuzzy logic, which fill a gap in the scene of the modeling and simulation techniques currently available to analyze heterogeneous complex systems. To test their effectiveness, such modeling frameworks have been employed to analyze the behavior of two real world systems, in the context of cellular biology. The results show that the developed frameworks correctly reproduce the behavior of complex systems and assess their response to perturbations, even when quantitative information is missing or some system components are not fully characterized. These frameworks could find applications in several fields, including, but not limited to, biology, medicine and pharmacology, which often face such challenges.

Acknowledgements

I would like to thank all the scientists and friends I met throughout these three years, for sharing their experiences and knowledge with me, but also for all the fun we had together. In particular, I would like to thank our collaborators at the BtBs Department, University of Milano-Bicocca, whose help contributed to part of the work presented in this thesis. Many thanks go to Prof. James Faeder and Sinan, for hosting me at the University of Pittsburgh, Pittsburgh, Pennsylvania, USA, where I spent many amazing and unforgettable days. Huge thanks to Andrea and Leonardo, for sharing with me the first two years of this adventure (and teaching me how to survive the department's bureaucracy). I also want to thank Prof. Giancarlo Mauri, whose support granted me the possibility to undertake this PhD program.

Many big thanks to Marco and Paolo, for being wonderful co-workers, mentors and friends, for helping me on so many occasions I cannot even remember, and for sharing countless funny moments together. The biggest "thank you" of all goes to Daniela, for being a strict mentor, for being a great advisor (even before the start of this PhD), for her friendship, and especially for teaching me how to be a better scientist.

I also want to thank my parents for supporting this crazy idea of mine, my "Aikido" friends and my dear friends from the "Sabbabbato" group for always being there for me. My final thanks go to my furry friend Mica, for her precious help during the writing, and Caro, for changing my life completely with her never-ending love and support.

Contents

List of Abbreviations

Abbreviation	Definition
API	Application Programming Interface
BDF	Backward Differentiation Formulae
CME	Chemical Master Equation
COG	Center Of Gravity
DFM	Dynamic Fuzzy Model
EM	Euler's Method
FCL	Fuzzy Control Language
FIS	Fuzzy Inference System
FML	Fuzzy Markup Language
FN	Fuzzy Inference Network
FUMOSO	FUzzy MOdels SimulatOr
FuzzX	Fuzzy-mechanistic modeling of compleX systems
GD	Gradient Descent
GUI	Graphical User Interface
HC	Hill Climbing
LSODA	Livermore Solver of Ordinary Differential Equations
MAK	Mass Action Kinetics
ODE	Ordinary Differential Equation
RBM	Rule-Based Model
RK4	Runge-Kutta 4
SA	Simulated Annealing
SSA	Stochastic Simulation Algorithm
TSK	Takagi-Sugeno-Kang

List of Symbols

Roman letters

Symbol	Definition
A	A generic fuzzy set
$a_j(\cdot)$	Propensity function of reaction P_j
c_j	Stochastic constant associated to the j-th reaction
$d_j(t)$	Distinct combinations of reactant molecules in reaction P_j at time t
e	Number of perturbed variables at each iteration in Simulated Annealing
$F(\cdot)$	Objective function
$\mathcal{F}$	Fuzzy module of FuzzX
f	Number of linguistic variables
$\mathcal{I}$	Interface of FuzzX
k	Number of fuzzy sets/linguistic terms in l
$\mathcal{L}$	Set of linguistic variables
$\widetilde{\mathcal{L}}$	Set of perturbable linguistic variables
l	Linguistic variable
M	Set of membership functions belonging to l
$\mathcal{M}$	Mechanistic module of FuzzX
m	Number of mechanistic variables/species (according to context)
N	Number of variables in a dynamic fuzzy model
$\widetilde{N}$	Number of perturbable variables in a dynamic fuzzy model
P	Mechanistic process/reaction (according to context)
$\mathcal{P}$	Set of mechanistic processes/reactions (according to context)
$\mathbb{P}(\cdot)$	Probability
p	Number of mechanistic processes/reactions (according to context)
q	Number of fuzzy rules
R	Fuzzy rule
$\mathcal{R}$	Set of fuzzy rules
$S(t)$	State of the dynamic fuzzy model at time t
$s_n(t)$	State of the n-th linguistic variable at time t
T	Temperature parameter of Simulated Annealing
t	Simulation time
t_{max}	Maximum simulation time
$U^{\mathtt{F}}$	Update function of fuzzy module
$U^{\mathtt{M}}$	Update function of mechanistic module

$\mathcal{U}$	Universe of discourse
u	Element of $\mathcal{U}$
V	Volume of a biochemical model
$\mathcal{V}$	Set of variables in a mechanistic model
$\mathcal{V}^{\mathtt{F}}$	Set of linguistic variables in the fuzzy module of FuzzX
$\mathcal{V}^{\mathtt{M}}$	Set of variables in the mechanistic module of FuzzX
v	Numerical value of a linguistic variable
w	Degree of satisfaction of a fuzzy rule
X	Name of a linguistic variable
$\mathbf{X}^{\mathtt{F}}(t)$	State of the fuzzy module at time t
$\mathbf{X}^{\mathtt{M}}(t)$	State of the mechanistic module at time t
$\mathcal{X}_M$	A set in which $x^{\mathtt{M}}$ assumes value
x	Variable of a mechanistic model
$x^{\mathtt{M}}$	Variable of the mechanistic module
$x^{\mathtt{F}}$	Variable of the fuzzy module
$\mathcal{Y}$	Initial state of a dynamic fuzzy model
y_n	Initial state of the n-th linguistic variable in a dynamic fuzzy model
Z	Union of output fuzzy subsets
z	Fuzzy set or function appearing in the consequent of a fuzzy rule

Greek letters

Symbol	Definition
α_{ij}	Stoichiometric coefficient of the i-th reactant species in the j-th reaction
β_{ij}	Stoichiometric coefficient of the i-th product species in the j-th reaction
Δ	Mechanistic module simulation time in FuzzX
δ	Sampling time for the programmed cell death dynamic fuzzy model
ϵ	Integration step
$\eta_e(\pi)$	Neighborhood function employed by Simulated Annealing in FUMOSO
θ	Set of parameters in a mechanistic model
$\theta^{\mathtt{M}}$	Set of parameters in the mechanistic module of FuzzX
κ	Number of simulation steps in FuzzX
Λ	Term set of l
λ	Linguistic term
$\mu_A(\cdot)$	Membership function of fuzzy set A
Π	Set of possible solutions to an optimization problem
π	Optimization problem solution/perturbation of a dynamic fuzzy model
ρ	Random number sampled from a uniform distribution
σ	Final output of a fuzzy inference
τ	Time step in stochastic simulation algorithm
$\phi(t)$	Input function in a dynamic fuzzy model
$\widetilde{\phi}(t)$	Perturbation function/perturbed state in a dynamic fuzzy model
Ω	Complex system

Chapter 1

Introduction

"Complex system" is often used as an umbrella term to cover a wide variety of natural or artificial structures, whose functioning cannot be outlined by a mere description of their constituting parts. Indeed, a common feature of all complex systems is that they display emergent dynamic behaviors, resulting from the physical or functional interactions existing among their components [1]. Given their ubiquity in nature, scientists in several fields of research developed different mathematical modeling tools and simulation techniques in order to grasp the inner working of such systems, often leading to relevant breakthroughs [2].

The reasons for such flourishing of modeling techniques can be found in the differences existing in the educational backgrounds and expertise of the scientists working on these systems [1, 3, 4]. However, the need of devising different modeling techniques stemmed also from other reasons: complex behaviors can emerge at multiple scales of spatial, temporal and functional organization, the nature of the available data can vary significantly, and the existing knowledge about the system can be incomplete. Even though they display many differences, each modeling technique aims at capturing distinct characteristics of the system under analysis and tackle open questions about it. The existing modeling approaches can be roughly partitioned in two main classes: *quantitative* (or mechanistic) and *qualitative* models [5, 6].

Mechanistic models are generally characterized by a high level of detail, they are based on numerical data, display the presence of parameters that control the dynamic behavior of the whole system, and are close to the physical reality of the processes that are being modeled. Common examples include approaches based on ordinary/partial differential equations [7], Monte Carlo and Markov chain methods [8], algebraic and agent-based modeling [9, 10]. On the contrary, the definition of qualitative models is based on uncertain or imprecise information, so that phenomena are described in a more approximate way with respect to the physical reality [11]. The approximations introduced by qualitative models, however, can bring several advantages: qualitative models are closer to human perception and natural language than their quantitative counterpart, making their understanding easier to researchers with basic or no expertise in mathematical modeling, and their simulation is generally computationally less demanding. Qualitative modeling includes

models based on Bayesian inference [12], Boolean logic [13] or multi-valued discrete logic [14, 15], fuzzy logic [16], and graph theory [17].

Among the existing qualitative modeling approaches, models based on fuzzy logic [18] had a huge impact in the last decades, finding applications across multiple research fields. Fuzzy logic was the first (and probably the most successful) attempt to formalize in a sound mathematical framework the way humans "compute" with words [18]: it is an extension of multi-valued logic, introduced by Lotfi A. Zadeh [19, 20], specifically designed to deal with approximate reasoning. Nowadays, its applications include pattern recognition, information retrieval, classification/regression and decision making [21, 22], but in its first years it became visible mostly because of its success in the fields of process control and modeling of complex systems [11]. One key feature that distinguishes fuzzy logic from classical logic is that fuzzy logic statements (or fuzzy rules) can be either true or false "to a certain degree". This peculiar characteristic was made possible by the introduction of fuzzy sets [19]. This particular extension of classical sets provides a means to effectively represent vague, non-crisp concepts, and describe sets whose boundaries are not well defined, which are often used in everyday spoken language (*e.g.*, when describing the height of a person, we can say that they are short, medium, tall or very tall). In addition to common language, vague or "fuzzy" quantities can be used to model imprecise knowledge regarding the system under analysis, or components that cannot be measured in a precise way. Such cases, which at first might appear very unlikely when dealing with "hard" sciences, can be common in several research areas: for example in medical studies or in cellular biology, where researchers deal with extremely intricate systems, like cells or whole organisms, and data is often available only for limited parts of the system and in qualitative form (*e.g.*, MRI images, gene expression studies). Moreover, one key advantage of fuzzy sets is their flexibility, which allows them to be suitable to represent components or properties of the systems whose nature can be highly heterogeneous (*e.g.*, not representable by means of real-valued variables, and/or spanning different orders of magnitude, and/or having units of measure of different nature). This characteristics of fuzzy sets can be exploited to integrate into one single framework different modeling approaches.

As a matter of fact, qualitative and quantitative modeling have been separately used to address different problems, in several research fields [5, 11, 23, 24]. Contacts between this two worlds have been limited, and efforts in trying to bridge them together have been restricted to domain-specific applications. For example, in [25, 26] the authors review and discuss several approaches that try to reconcile different mathematical formalisms with the aim of modeling cellular systems; the most notable one is the approach presented in [27], where the authors managed to simulate a whole cell belonging to one of the most simple living organisms, the bacterium *M. genitalium*. Other examples include modeling of the cardiovascular system by means of fuzzy inference systems and ordinary differential equations [28], methods for refining a qualitative model predictions with quantitative data [29, 30], and

mechanistic simulators with embedded methods that generate qualitative models to provide a causal explanation of the observed dynamics [31].

The existence of profound differences between the plethora of mathematical formalisms adopted to represent complex systems undoubtedly poses an additional challenge and hinders the process of their integration. However, as it can be seen from the aforementioned examples, complex systems generally show features that could benefit from a description that simultaneously exploits the advantages provided by both the qualitative and quantitative approaches. Indeed, adopting both approaches to represent different parts of a complex system might improve its overall representation and understanding. Moreover, precise and quantitative data is not always available or measurable for all system components and their mutual interactions. Thus, integrating uncertain and qualitative information with precise and quantitative data, whenever available, could result in the definition of more complete models able to take into account all the processes occurring in the system under analysis, and to uncover unexpected or yet unknown emergent behaviors in normal and perturbed conditions.

1.1 Aims and motivations

The aims of this work are twofold.

The first aim is to design and implement a novel and general-purpose computational method, based on fuzzy logic, to allow the modeling and analysis of heterogeneous complex systems. This new method can be used to define fuzzy models, that display the following features:

- they are dynamic models, able to simulate and predict the temporal evolution of the system in both unperturbed and perturbed conditions, although they do not require the availability of quantitative parameters and exact values of the state or the amount of the system components;
- they allow to represent heterogeneous components, spanning different orders of magnitude of spatial, temporal or functional organization, with a formalism that is close to natural language, and thus able to provide interpretable models;
- they can be coupled to optimization algorithms, in order to automatically identify sets of system components whose perturbation can maximize, or minimize, a desired behavior, and ultimately facilitate the control over the system.

Although several fuzzy logic tools and libraries are available in the literature, none of them was specifically designed to support the modeling and the dynamical simulation of heterogeneous systems, as well as the optimization of their behavior. For this reason, a novel software was developed to support the definition, editing, export and simulation of heterogeneous fuzzy models of complex dynamical systems. Moreover, an application of this novel method is also provided: a dynamic

fuzzy model of a complex, heterogeneous system consisting of oncogenic K-ras cancer cells, which was defined to study the mechanisms that drive cancer cells to death or survival.

The second aim of this thesis is to develop a novel framework, called FuzzX, able to model and simulate *hybrid models* of complex systems, consisting in both a qualitative module and a mechanistic module. In particular, the mechanistic module allows to accommodate mechanistic models formalized by means of any kind of fully parameterized modeling approach (*e.g.*, algebraic equations, ordinary differential equations, Markov jump processes), while the qualitative module is formalized as a fuzzy inference network. The final purpose of this framework is to bridge together qualitative and mechanistic models: it is designed to allow the communication and exchange of information between the two modules (by means of a common interface), in order to perform simulations in which both modules can influence each other's behavior. A software implementation of FuzzX is presented, as a new, general purpose simulator of hybrid models. Lastly, in order to show its potentiality, an application of this new framework is shown, where a mechanistic model of a biochemical signaling pathway in the yeast *S. cerevisiae*, characterized by complex non-linear behaviors, is redefined and analyzed in terms of a hybrid model. This application shows that FuzzX provides several advantages, including: the simplification of model complexity by means of a formalism close to human language, requiring a reduced number of free parameters; the possibility to run simulations of hybrid models, conveying both quantitative and qualitative information; the ability of hybrid models to reproduce the outcome of a detailed mechanistic model.

Together, these two contributions fill a gap in the scene of the modeling and simulation techniques currently available to analyze heterogeneous complex systems.

1.2 Overview of contents

This thesis is structured into two parts.

Part I contains prerequisites and basic notions that will be exploited in the development of this work. In particular, Chapter 2 provides an introduction to fuzzy set theory, fuzzy logic and fuzzy reasoning; Chapter 3 gives a brief description of optimization algorithms, with a focus on Simulated Annealing; Chapter 4 contains a description of the mechanistic modeling approaches employed in this work.

Part II presents the novel contributions developed during this thesis work. First, the description of dynamic fuzzy modeling and its coupling to optimization algorithms is given in Chapter 5; Chapter 6 presents a newly developed Python library to handle fuzzy logic, together with some practical examples; lastly, the description of a novel framework for hybrid modeling and simulation is given in Chapter 7. The applications of these methods to the modeling of cancer cell death and biochemical signaling pathways are provided in Chapter 8, together with a discussion

of the obtained results. Finally, in Chapter 9 some concluding remarks and future developments are presented and discussed.

The thesis also includes three appendices: Appendix A contains the fuzzy rules of the model defined in Section 8.1, while Appendix B provides information regarding the experimental procedures used for its validation; Appendix C presents the mechanistic model considered in Section 8.2.

1.3 Scientific production

The contents of this thesis are based on the following publications.

Journal papers

- Nobile M.S., Votta G., Palorini R., Spolaor S., De Vitto H., Cazzaniga P., Ricciardiello F., Mauri G., Alberghina L., Chiaradonna F., Besozzi D.
  Fuzzy modeling and global optimization to predict novel therapeutic targets in cancer cells
  Bioinformatics, 2019 [pending decision].

- Spolaor S., Nobile M.S., Mauri G., Cazzaniga P., Besozzi D.
  Coupling mechanistic approaches and fuzzy logic to model and simulate complex systems
  IEEE Transactions on Fuzzy Systems, 2019.

Conference proceedings

- Spolaor S.
  Bridging qualitative and quantitative modeling with FuzzX
  20th Italian Conference on Theoretical Computer Science (ICTCS2019), 2019 [in press].

Book chapters

- Spolaor S., Gribaudo M., Iacono M., Kadavy T., Oplatková Z.K., Mauri G., Pllana S., Senkerik R., Stojanovic N., Turunen E., Viktorin A., Vitabile S., Zamuda A., Nobile M.S.
  Towards human cell simulation
  High-Performance Modelling and Simulation for Big Data Applications. Springer, Cham, 2019. p. 221-249.

Posters and abstracts

- Spolaor S.
  Hybrid modeling of biological systems with FuzzX
  16th International Conference on Computational Intelligence methods for Bioinformatics and Biostatistics (CIBB2019), 2019, Bergamo, Italy
  [oral presentation].

- Cazzaniga P., Besozzi D., Nobile M.S., Spolaor S., Mauri G.
  A fuzzy logic based approach to handle cellular heterogeneity and the lack of quantitative parameters in dynamical models of biological systems
  11th Annual q-bio Conference, 2017, New Brunswick, NJ, USA.

- Besozzi D., Nobile M.S., Cazzaniga P., Spolaor S., Mauri G.
  Dealing with cellular heterogeneity and lack of quantitative parameters in dynamical modeling of biological systems: a fuzzy logic based approach
  CSHL Meeting "Cellular Dynamics & Models", 2017, Cold Spring Harbor, NY, USA.

Manuscript in preparation

- Spolaor S., Nobile M.S., Cazzaniga P., Besozzi D.
  Simpful: a simple fuzzy logic library

Additional work on related topics resulted in the following publications, which are not covered in this thesis.

Journal papers

- Besozzi D., Castelli M., Cazzaniga P., Manzoni L., Nobile M.S., Ruberto S., Rundo L., Spolaor S., Tangherloni A., Vanneschi L.
  Computational Intelligence for life sciences
  Fundamenta Informaticae, 2020, 171: 1-4.

- Nobile M.S., Vlachou T., Spolaor S., Bossi D., Cazzaniga P., Lanfrancone L., Mauri G., Pelicci P.G., Besozzi D.
  Modeling cell proliferation in human acute myeloid leukemia xenografts
  Bioinformatics, 2019, 35.18: 3378–3386.

- Tangherloni A., Spolaor S., Cazzaniga P., Daniela Besozzi D., Rundo L., Mauri G., Nobile M.S.
  Biochemical parameter estimation vs. benchmark functions: a comparative study of optimization performance and representation design
  Applied Soft Computing, 2019, 81: 105494.

- Tangherloni A., Spolaor S., Rundo L., Nobile M.S., Cazzaniga P. Mauri G., Liò P., Merelli I., Besozzi D.
  GenHap: a novel computational method based on genetic algorithms for haplotype assembly
  BMC Bioinformatics, 2019, 20.4: 172.

- Nobile M.S., Porreca A.E., Spolaor S., Manzoni L., Cazzaniga P., Mauri G., Besozzi D.
  Efficient simulation of reaction systems on Graphics Processing Units
  Fundamenta Informaticae, 2017, 154.1-4: 307-321.

Conference proceedings

- Spolaor S., Fuchs C., Kaymak U., Nobile M.S.
  A novel multi-objective approach to fuzzy clustering
  2019 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2019 [accepted].

- Nobile M.S., Vlachou T., Spolaor S., Cazzaniga P., Mauri G., Pelicci P.G., Besozzi, D.
  ProCell: investigating cell proliferation with swarm intelligence
  2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). IEEE, 2019. p. 1-8.

- Fuchs C., Spolaor S., Nobile M.S., Kaymak U.
  A swarm intelligence approach to avoid local optima in fuzzy c-means clustering
  2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE, 2019. p. 1-6.

- Tangherloni A., Rundo L., Spolaor S., Nobile M.S., Merelli I., Besozzi D., Mauri G., Cazzaniga P., Liò P.
  High performance computing for haplotyping: models and platforms
  European Conference on Parallel Processing. Springer, Cham, 2018. p. 650-661.

- Nobile M.S., Tangherloni A., Rundo L., Spolaor S., Besozzi D., Mauri G., Cazzaniga P.
  Computational intelligence for parameter estimation of biochemical systems
  2018 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2018. p. 1-8.

- Tangherloni A., Rundo, L., Spolaor, S., Cazzaniga, P., Nobile, M. S.
  GPU-powered multi-swarm parameter estimation of biological systems: a master-slave approach
  2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP). IEEE, 2018. p. 698-705.

- Spolaor S., Tangherloni A., Rundo L., Nobile M.S., Cazzaniga P.
  Estimation of kinetic reaction constants: exploiting reboot strategies to improve PSO's performance
  International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics. Springer, Cham, 2017. p. 92-102.

- Spolaor S., Tangherloni A., Rundo L., Nobile M.S., Cazzaniga P.
  Reboot strategies for Particle Swarm Optimization and their impact on parameter estimation
  2017 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). IEEE, 2017. p. 1-8.

Posters and abstracts

- Nobile M.S., Vlachou T., Spolaor S., Bossi D., Cazzaniga P., Lanfrancone L., Besozzi D., Pelicci P.G., Mauri G.
  A cell proliferation model of human acute myeloid leukemia xenograft
  26th Conference on Intelligent Systems for Molecular Biology (ISMB), 2018, Chicago, IL, USA.

- Tangherloni A., Spolaor S., Rundo L., Nobile M.S., Cazzaniga P., Liò P., Merelli I., Besozzi D., Mauri G.
  GenHap: haplotype assembly using genetic algorithms
  15th BITS meeting, 2018, Turin, Italy.

- Tangherloni A., Spolaor S., Rundo L., Nobile M.S., Merelli I., Cazzaniga P., Besozzi D., Mauri G., Liò P.
  GenHap: a novel computational method based on genetic algorithms for haplotype assembly
  Extended abstract, Methods, tools & platforms for Personalized Medicine in the Big Data Era, NETTAB Workshop, 2017, Palermo, Italy.

- Nobile M.S., Cazzaniga P., Tangherloni A., Spolaor S., Besozzi D., Mauri G.
  High-performance computing in Systems Biology: accelerating the simulation and analysis of large and complex biological systems
  17th International Conference on Systems Biology (ICSB), 2016, Barcelona, Spain.

Part I

Theoretical background

Chapter 2

Fuzzy logic

L. A. Zadeh used to describe fuzzy logic as the act of computing with words [18]. Indeed, the primary aim of fuzzy logic is representing the mechanisms underlying approximate modes of reasoning, that are common in our everyday language. Fuzzy logic is a mathematical framework that can be exploited to connect the qualitative world of words to the quantitative world of measures. The numerous advantages that fuzzy logic is able to provide stem from this characteristic. Fuzzy logic can be applied to increase the interpretability of computational models and/or to provide cost-effective approximations, even in the absence of precise quantitative information. Moreover, the flexibility of fuzzy sets allows to model a wide variety of phenomena, including heterogeneous entities and concepts that are not amenable of precise measurement. This flexibility can also be employed to "fuzzify" existing mathematical frameworks and/or to connect them, in order to deal with multiple sources of uncertainty. The following sections will briefly describe some elements of fuzzy logic, essential to the understanding of the novel work presented in this thesis.

2.1 Fuzzy sets

In classic set theory, an object can either belong or not to a given set. Intuitively, we perceive these sets as provided with well-defined, crisp borders: for example, "dog" and "cat" undeniably belong to the set of "animals". A fuzzy set represents an extension of the ordinary set [19]: its boundaries are not sharp, so that elements can belong to it to "some degree". Fuzzy sets allow to represent concepts of human reasoning that cannot be represented by crisp sets: for example, the set of "tall people" may contain people that are undoubtedly tall (*i.e.*, with a maximum degree of "membership" to the set), but also people that are not as tall, but definitely not short (*i.e.*, with a lower degree of membership).

Formally, given a universe of discourse $\mathcal{U}$, and u as its generic element, a fuzzy set A is characterized by a membership function, denoted by $\mu_A : \mathcal{U} \rightarrow [0, 1]$, which maps an element u to its degree of membership; in other words, $\mu_A(u)$ is the degree of membership of u to A. If A is an ordinary set, then its membership function $\mu_A(u)$ will assume only values 1 or 0. An example of this concept is given in Figure 2.1. On the left side, two crisp sets are used to partition the temperature of water

into two sets, "cold" and "hot", and a clear threshold separate what can be defined as cold or hot water; on the right, side instead, three fuzzy sets are employed to partition the same set of temperatures. The fuzzy partition allows to have a smooth transition from "cold" to "warm" water, as well as from "warm" to "hot" water, in a continuous way, without a clear, crisp threshold. The right panel in Figure 2.1 therefore shows an example of a representation that is closer to human perception.

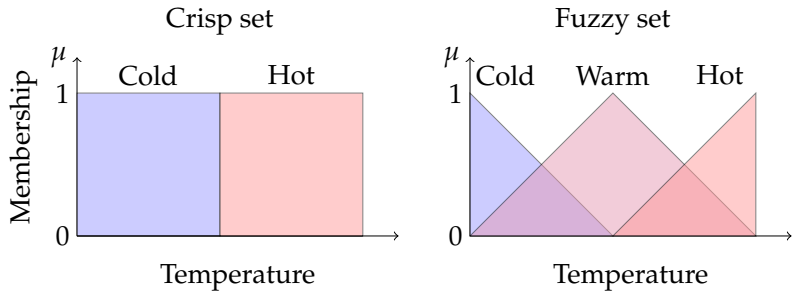


FIGURE 2.1: Comparison between crisp sets (left) and fuzzy sets (right) used to represent water temperature.

Several definition and properties of fuzzy sets are also direct extensions of the corresponding ones for ordinary sets:

- the fuzzy set A is empty if and only if $\mu_A(u) = 0, \forall u \in \mathcal{U}$;
- two fuzzy sets A and B are equal if and only if $\mu_A(u) = \mu_B(u), \forall u \in \mathcal{U}$;
- the complement of A, denoted by $\overline{A}$, is defined by $\mu_{\overline{A}}(u) = 1 - \mu_A(u)$;
- the fuzzy set A is contained in B if and only if $\mu_A(u) \leq \mu_B(u), \forall u \in \mathcal{U}$;
- the union of fuzzy sets A and B is a fuzzy set C, defined by $\mu_C(u) = \max(\mu_A(u), \mu_B(u)), u \in \mathcal{U}$;
- the intersection of fuzzy sets A and B is a fuzzy set C, defined by $\mu_C(u) = \min(\mu_A(u), \mu_B(u)), u \in \mathcal{U}$.

Extension of additional properties (De Morgan laws, distributive laws) and additional operations on fuzzy sets can be found in [19].

It should be noted that fuzzy sets represent a form of vagueness and uncertainty that is typical of human language, but that is not related to uncertainty derived by randomness, as fuzzy sets do not represent probability distributions [19]. As suggested in [32], a degree of membership represented by a fuzzy set can be interpreted, depending on its application, in three ways:

- as a degree of similarity to a prototype object/condition. The most common interpretation, it is used in clustering, regression analysis but also control theory and modeling;

- as a degree of preference, representing an intensity of preference in favor of a particular object, or the feasibility of selecting a certain value in a decision task. This view is commonly adopted in the fields of optimization or decision making;
- as a degree of uncertainty, inside the framework of possibility theory [33], adopted in approximate reasoning.

When applying fuzzy sets, it is advisable to identify a correct interpretation, which has to be common among all fuzzy sets. Since this work deals with modeling, all the fuzzy sets presented in the following sections will fall under the first interpretation, that is, they describe degrees of similarity to a prototypical condition (*e.g.*, the state of a system component).

It should also be noted that higher-order fuzzy sets, commonly referred to as type-2 fuzzy sets exist [34]. These fuzzy sets are characterized by membership functions that are fuzzy as well, thus they feature more degrees of freedom and they can model additional layers of uncertainty. Since all the works presented here employ classic, type-1 fuzzy sets, this thesis will not deal with type-2 fuzzy sets.

2.2 Linguistic variables

The definition of fuzzy sets allows to introduce another fundamental concept in fuzzy logic, that is, linguistic variables. Differently from common numerical variables, a linguistic variable can assume as values words or sentences in some natural or artificial language [35–37]. Linguistic variables effectively connect the quantitative world of measures, to the qualitative world of human reasoning. In fact, they provide a means to approximate phenomena that are too complex or too ill-defined, so that they can be amenable of a description in conventional quantitative terms. This is possible thanks to fuzzy sets, which represent broad collections of elements inside a universe of discourse, analogously to the role played by words in a natural language. For example, referring again to Figure 2.1, the linguistic variable "water temperature" can be represented by partitioning the universe of discourse into three fuzzy sets, which roughly describe what could be represented as a precise and quantitative value of temperature as "cold", "warm" and "hot" terms, according to human perception.

Formally, in this work a linguistic variable l is defined as the quadruple

$$l = \{X, \mathcal{U}, \Lambda, M\},$$

where:

- X is the name of the linguistic variable;
- $\mathcal{U}$ denotes the universe of discourse in which X has meaning;

- $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ is the term set of X, that is, the set of k linguistic values that l can assume;
- $M = \{\mu_{\lambda_1}, \ldots, \mu_{\lambda_k}\}$ is the set of fuzzy subsets in which U is partitioned. Each term $\lambda \in \Lambda$ is associated with one and only one fuzzy set $\mu \in M$.

In order to avoid an excess of notation, it should be pointed out that: (*i*) since a fuzzy set is defined by its membership function, μ will be used to denote both the fuzzy set and its membership function; (*ii*) we can refer to l just by using its name X.

Worthy of note, in the original definition of linguistic variables [36], L. A. Zadeh took into account the presence of linguistic hedges, such as "very", "more", "less", able to modify the membership values of the linguistic terms. It should be pointed out that this work does not deal with linguistic hedges, and whenever modifiers such as "very" or "more" appear inside fuzzy models, they do appear only as part of different linguistic terms.

2.3 Fuzzy rules

IF/THEN rules are commonly employed in classic logic to express implication and represent "pieces of knowledge". They can be extended into fuzzy IF/THEN rules for the purpose of fuzzy set-based approximate reasoning [38]. Fuzzy IF/THEN rules are logic rules in which the antecedent, or both antecedent and consequent, are fuzzy sets rather than crisp. Although different interpretations of fuzzy implication can be found in the literature (see, *e.g.*, [38]), they are generally defined by an expression in the form:

$$\text{IF } X \text{ IS } \lambda \text{ THEN } Y \text{ IS } z, \tag{2.1}$$

where X and Y are the names of two linguistic variables, λ is a term of X and z is either a crisp value, a term of Y, or a function of the variables appearing in the antecedent. In general, the antecedent of a fuzzy rule can be any well-formed logic expression, formed by means of the operators AND, OR, NOT. For example,

$$\text{IF } X \text{ IS } \lambda \text{ AND } X' \text{ IS } \lambda' \text{ THEN } Y \text{ IS } z.$$

The degree of satisfaction w is defined as the degree to which the given values of the input linguistic variables match the antecedent of a fuzzy rule. In particular, for rules in the form given by Expression 2.1, given $v^X \in \mathcal{U}^X$ as the value of X, then $w = \mu_\lambda(v^X)$. In general, for rules whose antecedent is a well-formed logic expression, w can be computed by applying each operator (in order of appearance) as follows. Given X and X' as two linguistic variables, λ and λ' has their linguistic terms, and v^X and $v^{X'}$ as their values:

- X IS NOT λ corresponds to the complement of λ: $w = 1 - \mu_\lambda(v^X)$;
- X IS λ AND X' IS λ' corresponds to the intersection of λ and λ': $w = \min(\mu_\lambda(v^X), \mu_{\lambda'}(v^{X'}))$;

- X IS λ OR X' IS λ' corresponds to the union of λ and λ':
  $w = \max(\mu_\lambda(v^X), \mu_{\lambda'}(v^{X'}))$.

The complete set of fuzzy rules is known as a "fuzzy rule base".

Different definitions of the logic operators, implication and fuzzy rules exist; these gave rise to a vast body of literature and, ultimately, affect the conclusions that are drawn by a fuzzy rule base in fuzzy reasoning. This chapter will focus in particular on the two most common and successful interpretations of fuzzy rules and fuzzy inference: the Mamdani method, and the Takagi Sugeno Kang (also known as TSK or Sugeno) method. In both methods, fuzzy rules provide a rough description of the relation between their input (*i.e.*, appearing in the antecedent) and output (*i.e.*, appearing in the consequent) variables. In fact, fuzzy antecedents in fuzzy rules provide the basis for an interpolation mechanism, as explained in the next section.

2.4 Fuzzy inference systems

Linguistic variables and fuzzy rules represent the foundations of Fuzzy Inference Systems (FISs) [39]. Fuzzy inference can be described as the process of mapping a given input to an output by means of fuzzy logic. FISs found successful applications in several fields, ranging from automatic control and expert systems, to classification, regression analysis, computer vision etc. Because of this strong multidisciplinarity, they are known by different names, such as fuzzy logic controllers, fuzzy systems, fuzzy-rule-based systems, fuzzy expert systems, and fuzzy models. This success was also due to the fact that, as neural networks, it was proven that a broad number of FISs are universal approximators [40–42], that is, they are capable of approximating any real continuous function on a compact set with an arbitrary accuracy. Notably, among these there are the Sugeno FISs, commonly employed in control and modeling.

In general, a common architecture of FISs (depicted in Figure 2.2) can be described as follows:

1. fuzzification of crisp inputs into their linguistic variables and evaluation of their membership degree by means of the membership functions;

2. inference of rule conclusions (*i.e.*, rule evaluations) and their aggregation into a single output;

3. defuzzification (optional) of the fuzzy rules' output into a single crisp value.

The fuzzy knowledge base, that is, the set of linguistic variables together with their fuzzy sets and fuzzy rules, is exploited in each passage of the inference process. Initially, FISs were designed and developed mainly for automatic control purposes: the earliest example being [43], where the authors developed what is now known as the Mamdani inference method to control a steam engine. In this setting, the knowledge base is generally built by domain experts. However, nowadays there exist a number

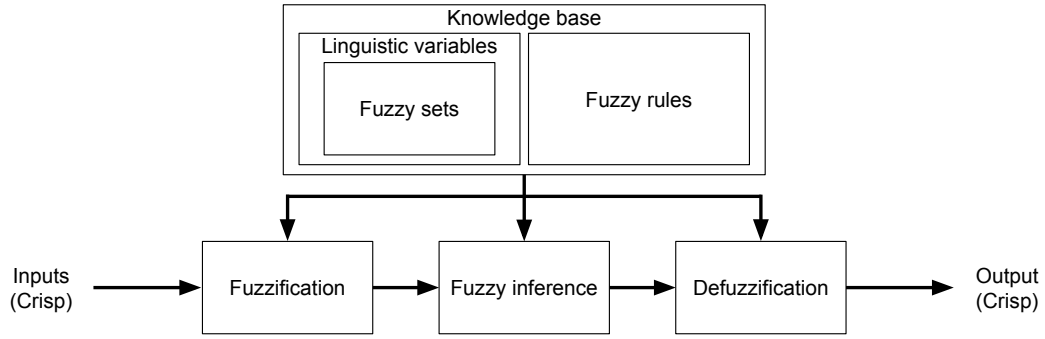


FIGURE 2.2: Graphical representation of the general architecture of a fuzzy inference system.

of applications in which the knowledge base is automatically built starting from a given data set (for example classification, regression analysis, etc. [44]). This thesis work will focus on modeling and simulation of complex systems, thus all knowledge bases presented are to be considered as expert-defined. Some remarks on possible data driven applications and future developments on the automatic inference of fuzzy rule bases will be given in Chapter 9.

The Mamdani [43] and the Sugeno [45] inference methods—the two most common and widely accepted fuzzy inference methods, especially in the field of control and fuzzy modeling—share the general architecture described above, although they differ in the definition of the consequents of the fuzzy rules and how they aggregate the rules evaluations. These two inference methods are described in detail in the following subsections.

2.4.1 Mamdani fuzzy inference systems

The Mamdani method [43] was the first proposed method of fuzzy inference. Mamdani type fuzzy rules are characterized by having fuzzy sets as consequents: thus, the conclusions drawn after a rule evaluation must be combined into a single output, which can then be defuzzified.

The most common Mamdani inference method—adopting *min* as implication operator, *max* as aggregation operator and *center of gravity* (COG) as defuzzification method—can formally be defined as follows. Let $\mathcal{L} = \{l_1, \ldots, l_f\}$ be a set of input linguistic variables, with $v_1, \ldots, v_f$ being their values, and $\mathcal{R} = \{R_1, \ldots, R_q\}$ be a set of fuzzy rules, where $R_j \in \mathcal{R}$, $j = 1, \ldots, q$, is defined as in Expression 2.1 and has as antecedent a well-formed logic expression composed of variables belonging to $\mathcal{L}$. Additionally, let Y be the name of the output variable and z a linguistic term of Y. To obtain the final output value σ of Y, the Mamdani inference method proceeds as follows:

- fuzzify the values $v_1, \ldots, v_f$ into their respective membership functions;
- calculate the degree of satisfaction w_j of each rule R_j, as described in Section 2.3;

- for each rule, identify the fuzzy subset z_j^w of z defined by the function $\min(w, \mu_z)$;
- calculate the output fuzzy set $Z = \bigcup_{j=1}^{q} z_j^w$;
- calculate the defuzzified output as the center of the area covered by $\sigma = COG(Z)$.

An example of Mamdani inference is depicted in Figure 2.3.

Mamdani FISs are generally considered easy to understand and characterized by more interpretable rules, since the consequents are defined by fuzzy sets. However, their performance is strongly dependent on the aggregation and defuzzification methods, and they give unreliable results when many rules are firing together (*e.g.*, the outputs converge towards the center of the universe of discourse).

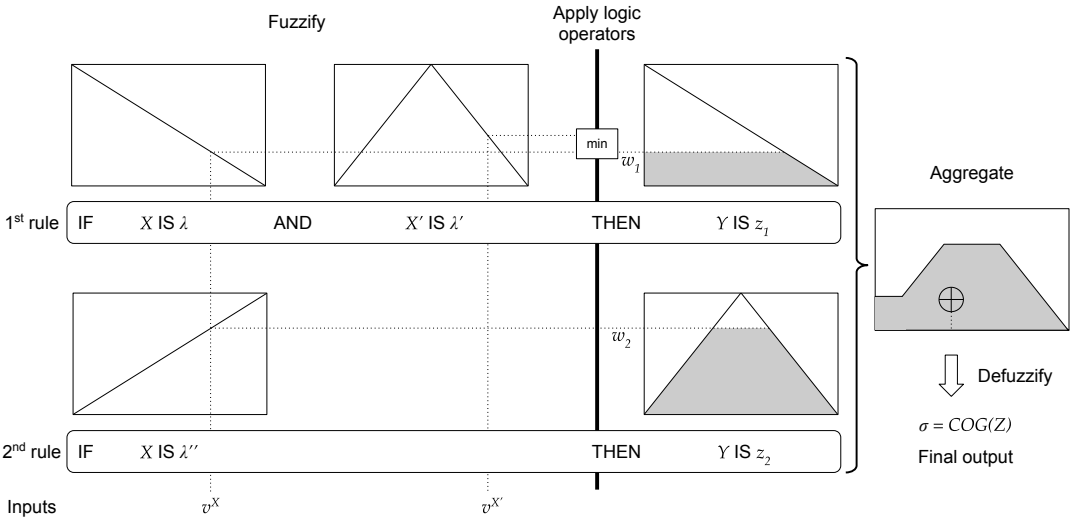


FIGURE 2.3: Graphical representation of an example of Mamdani inference, consisting of two input linguistic variables and two fuzzy rules.

2.4.2 Sugeno fuzzy inference systems

The Sugeno inference method [45] is characterized by the presence of fuzzy rules in which the consequent is a function of the variables appearing in the antecedent. Contrary to the Mamdani method, since the rules' evaluation returns crisp values, there is no need for defuzzification.

The Sugeno inference is formally defined as follows. Let $\mathcal{L} = \{l_1, \ldots, l_f\}$ be a set of input linguistic variables, with $v_1, \ldots, v_f$ being their values, and $\mathcal{R} = \{R_1, \ldots, R_q\}$ be a set of fuzzy rules, where $R_j \in \mathcal{R}$, $j = 1, \ldots, q$, is in the form

$$\text{IF } X \text{ IS } \lambda \text{ AND } X' \text{ IS } \lambda' \text{ THEN } Y \text{ IS } z_j,$$

where the antecedent can be any well-formed logic expression composed of variables belonging to $\mathcal{L}$, Y is the name of the output variable and z_j is a function of the

linguistic variables appearing in the antecedent. To obtain the final output value σ of Y, the Sugeno inference method proceeds as follows:

- fuzzify the values $v_1, \ldots, v_f$ into their respective membership functions;
- calculate the degree of satisfaction w_j of each rule R_j, as described in Section 2.3;
- for each rule, compute the value of z_j;
- calculate the final output as $\sigma = \sum_{j=1}^{q} \frac{w_j z_j}{w_j}$.

An example of Sugeno inference is described in Figure 2.4.

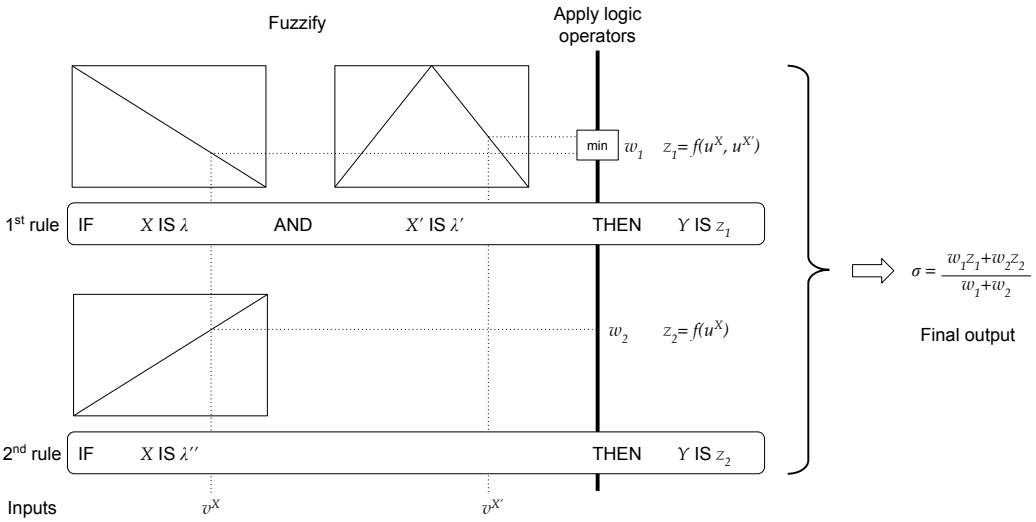


FIGURE 2.4: Graphical representation of an example of Sugeno inference, consisting of two input linguistic variables and two fuzzy rules.

Sugeno FISs are considered less interpretable, due to the fact that the consequents are in the form of functions. This is particularly true if higher-order functions are adopted. However, Sugeno FISs display several advantages: their output surface was proven to be continuous [46], thus they are more prone to mathematical analysis, and they are computationally less demanding compared to Mamdani FISs. Sugeno FISs are generally classified according to the order of the functions expressed in the consequent of the fuzzy rules. In particular, 0-order Sugeno FISs are characterized by the presence of a constant function (*i.e.*, a crisp value) in the consequent of the fuzzy rules. These FISs conserve all the advantages of higher-order Sugeno FISs, although they remain more interpretable to domain experts with respect to their higher-order counterparts.

2.5 Fuzzy networks

The definition of a multiple-input, single-output FIS might not be sufficient for several practical applications, especially if the variables to be modeled are strongly interdependent. To overcome this limitation, different formalisms were devised to efficiently "connect" more than one FIS, and, in particular, to feed the output of a FIS as input to a downstream one. Among these, one can find hierarchical FISs [47], multi-layer FISs [48], and fuzzy inference networks [49]. Fuzzy inference networks, or in short fuzzy networks (FNs), are the most versatile form of interacting FISs. A FN can be depicted as a directed graph, where nodes represent linguistic variables, and arcs the presence of some fuzzy rules governing them. Thus, connections between the nodes represent interactions in the form of rule outputs fed as variable inputs to a downstream FIS. These networks can be defined with arbitrary topologies, including cycles and feedback loops, a feature that proves particularly useful to model the non-linear interactions existing in complex systems [49–52].

Chapter 3

Global optimization

An optimization problem consists in finding, among the set Π of all possible solutions, the optimal solution π able to maximize (or minimize) the value of a given objective function $F(\cdot)$. Stated otherwise, in the case of maximization problems, it means finding $\pi \in \Pi$ such that $F(\pi) \geq F(\pi')$ for any $\pi' \in \Pi$. Optimization problems are generally classified in discrete or continuous, according to the nature of the search space of the possible solutions. When the size of the search space is large, a complete enumeration of the solutions becomes clearly impractical; several computational strategies exist, to explore the search space in a smart way, in order to determine the optimal solution or a close approximation of it.

3.1 Local and global optimization methods

Optimization methods are commonly divided in local and global optimization techniques. Local optimization includes the methods that improve the found solution only in its neighborhood of the search space, until they converge to a (local) optimum. Typical methods belonging to this class are Hill Climbing (HC) and Gradient Descent (GD) algorithms [53]. HC and GD start by choosing an initial putative solution π, usually on a random basis, exploiting some probability distribution, or any *a priori* knowledge about the search space. Then, at each iteration, they adopt a greedy approach in order to accept only new solutions that improve the value $F(\pi)$, until no further improvement is possible. A major drawback of local methods is that they can reach only the locally optimal solution of the basin of attraction in which they are initialized. Hence, they do not provide any guarantee that the best solution will be found. Although several variations exist to tackle this limitations (*e.g.*, multi-start strategies), they are not suited to optimize objective functions characterized by noisy, non-convex or multi-modal search spaces.

When dealing with a high number of local minima, global optimization techniques are preferred: these methods perform an effective and efficient exploration across the whole search space, increasing the chances of converging to the global optimum. Global optimization methods include several heuristics and metaheuristics methods (based, for example, on evolutionary computation or swarm intelligence), which, despite being approximate methods, become especially performing when

facing analytically intractable problems [54]. One of the most widespread method of global optimization is Simulated Annealing.

3.2 Simulated annealing

Simulated Annealing (SA) is a meta-heuristic, stochastic global optimization method, adapted from the Metropolis-Hastings sampling algorithm [55]. SA draws inspiration from the annealing in metallurgy, a process in which a metal is heated and then undergoes a controlled cooling in order to modify its crystal structure. A key feature of SA is that it can accept, with a given probability, worse solutions during the optimization process. This allows SA to escape local minima, switching between an exploration of the search space and the exploitation of the neighborhood of the found solution.

SA proceeds as follows. Starting from an initial solution π, at each iteration of the algorithm a new solution π' is created by perturbing π (*i.e.*, by sampling its neighborhood); the generation of π' is done accordingly to the structure of the solutions to the optimization problem. Then, given $F(\pi)$ and $F(\pi')$ as the goodness of fit of the previous and the new solutions, respectively, in the case of maximization problems π' is accepted with a probability:

$$\mathbb{P}(\text{accept } \pi' | F(\pi), F(\pi')) = \begin{cases} 1 & \text{if } F(\pi') > F(\pi) \\ \exp(\frac{-(F(\pi_i') - F(\pi_i))}{T}) & \text{otherwise,} \end{cases} \tag{3.1}$$

where T is the "temperature" hyper-parameter, affecting the exploration capability of SA. This process iterates until a termination criterion is met (most commonly, a maximum number of iterations). The output produced by SA is the solution π characterized by the best value $F(\pi)$.

The parameter T deeply affects the performance of SA; usually, the temperature starts from an initial value T_0 and linearly decreases to 0 during the iterations, progressively reducing the exploration capabilities of SA in favor of an exploitative behavior towards the end of the optimization. Since SA is a stochastic algorithm, different runs may yield different solutions; nevertheless, it is expected that SA will converge more often to optimal solutions, representing the attractors of the search space, as it was proven to asymptotically converge to the global optimum [56]. Among the advantages of SA, it should be noted that, differently from some local methods (such as GD, which exploits the gradient of $F(\cdot)$ in π), it does not require the objective function to be differentiable, and thus it is commonly used in discrete optimization problems.

Chapter 4

Mechanistic modeling and simulation

Mechanistic models have a key role in our understanding of natural phenomena and in the design of artificial systems. These models allow to integrate in a systematic manner the available data and help in the formulation of testable hypothesis, facilitating the analysis of complex systems and their behaviors. A relevant feature of these models is that they describe the mechanisms underlying the functioning of such systems and allow to simulate their evolution over time, by adopting mathematical formalisms whose representation is close to the physical reality.

Although several formalisms exists to represent the different nuisances of each system, they all share a general structure. Given a complex system Ω, a mechanistic model of Ω is characterized by a set $\mathcal{V}$ of variables, representing the components of the system, a set $\mathcal{P}$ of processes, describing the interactions existing between the components and how the system's dynamics evolve, and a set θ of parameters regulating such processes. The following sections will provide a brief description of the mechanistic modeling and simulation approaches exploited in this work, which focus on the simulation of biochemical systems.

4.1 Reaction-based models

A reaction-based model (RBM) [57] is a common formalism employed to model biochemical systems. Given a biochemical system Ω, a RBM is defined by a set $\mathcal{P} = \{P_1, \ldots, P_p\}$ of reactions, involving a set of chemical species $\mathcal{V} = \{x_1, \ldots, x_m\}$, and a set of parameters $\theta = \{c_1, \ldots, c_p\}$. A species variable $x_i \in \mathcal{V}$, with $i = 1, \ldots, m$, can assume values in $\mathbb{N}$ if the variables of the model represent number of molecules, or in $\mathbb{R}^+$ if they represent concentrations in a given volume V. A reaction $P_j \in \mathcal{P}$, with $j = 1, \ldots, p$, has the general following form:

$$P_j = \sum_{i=1}^{m} \alpha_{ij} x_i \xrightarrow{c_j} \sum_{i=1}^{m} \beta_{ij} x_i, \tag{4.1}$$

where $\alpha_{ij}, \beta_{ij} \in \mathbb{N}$ are the stoichiometric coefficients, that is, the number of molecules of species x_i involved in the reaction P_j as reactants and products, respectively. Finally, $c_j \in \theta$ represents a kinetic parameter, a constant that can assume values in $\mathbb{R}^+$ and encompasses the physical and chemical properties of the reaction.

Additionally, let the vector $\mathbf{X}(t) = (x_1(t), \ldots, x_m(t))$ describe the state of the m species at time t. $\mathbf{X}_0$ denotes the initial state of the system Ω represented by the RBM, that is, the state of the system $\mathbf{X}(t_0)$ at time t_0. Lastly, for each reaction P_j, a state change vector $\boldsymbol{\nu}_j = (\nu_{j1}, \ldots, \nu_{jm})$ can be defined, consisting of the elements $\nu_{ji} = \beta_{ij} - \alpha_{ij}$, with $\nu_{ji} \in \mathbb{Z}$, which represent the change in the amount of species x_i due to reaction P_j.

Biochemical reactions defined in RBMs are generally assumed to happen in a well-stirred, fixed volume at thermal equilibrium, and to obey the mass-action kinetics (MAK) [58, 59]. MAK is an empirical law stating that, in a well-stirred solution, the rate of an elementary reaction (*i.e.*, a reaction consisting in a single mechanistic step) is proportional to the product of the concentration of its reactants, raised to the power of the corresponding stoichiometric coefficients. Under this assumption, the temporal evolution of a RBM can be simulated by means of both deterministic or stochastic approaches. Common approaches include defining and numerically integrating the associated system of ordinary differential equations (deterministic), or sampling trajectories from the associated Chemical Master Equation (stochastic).

4.2 Ordinary differential equations

Traditionally, the temporal evolution of complex systems is modeled by means of a system of coupled ordinary differential equations (ODEs). Biochemical systems make no exception, and systems of coupled ODEs are often used to describe the rate of change of the species concentrations.

In particular, an RBM defined as in Section 4.1 can be represented as a system of ODEs as follows. Given $x_i(t)$ as the concentration of the i-th species at time t and p reactions in the form of Equation 4.1, by assuming MAK the system of coupled ODEs can be written as:

$$\frac{dx_i(t)}{dt} = \sum_{j=1}^{p} \nu_{ji}\, c_j\, \left(\prod_{i=1}^{m} x_i(t)^{\alpha_{ij}}\right), \text{ for } i = 1, \ldots, m.$$

It is worth noting that ODE based models are not limited to MAK, but can exploit different kinetics, depending on the context of their application (*e.g.*, in the case of biochemical systems, Hill functions and Michaelis-Menten kinetics are frequently used [58]).

ODE based models describe systems whose components can be represented with continuous variables and whose behaviors can be described by deterministic dynamics. If stochastic phenomena do play a role in the emergence of the system

behaviors (*e.g.*, the system shows bistability), stochastic modeling and simulation approaches are preferred (see Sections 4.4 and 4.5).

4.3 Numerical integration algorithms

Given an initial condition of Ω, the dynamics of the corresponding model can be obtained by solving the system of ODEs. Since obtaining the exact solution of such systems is not always analytically possible, numerical integration algorithms are commonly employed to obtain approximate solutions.

The simplest numerical integration algorithm is known as Euler's Method (EM) [60]. EM is an iterative algorithm that exploits, at regular points, the slope of the tangent to the unknown solution of the ODE system to approximate it. Precisely, let $y'(t) = f(t, y(t))$ be a generic ODE, $y(t_0)$ be an initial point, and $[t_0, t_{end}]$ be the time interval in which the approximate solution has to be determined. Given a step size ϵ, it is possible to define a set of γ time instants $t_0, \ldots, t_\gamma = t_{end}$, such that $t_n = t_0 + n\epsilon$, $n = 1, \ldots, \gamma$, and $\epsilon = \frac{t_{end} - t_0}{\gamma}$. Starting from t_0, at each step the unknown curve can be approximated as:

$$y_{n+1} = y_n + \epsilon f(t_n, y_n), \ \forall n = 0, \ldots, \gamma - 1$$

where $y_n = y(t_n)$. EM is a first-order, single step method, whose global error at a given time t is proportional to ϵ.

The idea underlying the EM often serves as the basis to construct more complex methods, such as the Runge-Kutta methods [60]. The Runge-Kutta methods are a family of iterative algorithms that improve the approximation provided by EM. The most popular Runge-Kutta algorithm is the fourth order Runge-Kutta method, or RK4. RK4 exploits a weighted average of four values at each integration step to calculate the solution. Specifically, RK4 calculates each new point y_{n+1} as

$$y_{n+1} = y_n + \frac{\epsilon}{6}(h_1 + 2h_2 + 2h_3 + h_4),$$

where h_1, h_2, h_3 and h_4 are calculated as

$$\begin{aligned} h_1 &= f(t_n, y_n), \\ h_2 &= f(t_n + \frac{\epsilon}{2}, y_n + \frac{\epsilon}{2} h_1), \\ h_3 &= f(t_n + \frac{\epsilon}{2}, y_n + \frac{\epsilon}{2} h_2), \\ h_4 &= f(t_n + \epsilon, y_n + \epsilon h_3). \end{aligned}$$

Being RK4 a fourth order method, its global approximation error at any given time is proportional to ϵ^4, resulting in an improved solution with respect to EM.

Although widely used, the RK4 method might be not efficient in solving systems of ODEs characterized by stiffness. Informally, stiffness can be described as the

phenomenon by which highly stable ODE systems show a high degree of instability when approximated by classical numerical methods [60]. A way to cope with this problem is reducing the step size of integration algorithms, but this might lead to a considerable increase in simulation time; additionally, in some ODE systems this instability leads to a significant propagation of approximation errors, eventually achieving poor solutions [61]. Several ODE systems of practical importance in scientific modeling exhibit stiffness, and over the years many methods were developed in order to cope with this phenomenon. One of the most efficient algorithms up to date for the numerical integration of stiff and non-stiff systems of ODEs is the Livermore Solver of Ordinary Differential Equations (LSODA) [62]. LSODA is an ODE solver that automatically detects the presence of stiffness during the execution, and dynamically switches to the class of integration methods that is likely to be more efficient on that part of the problem. In particular, LSODA adopts two families of integration algorithms: a class of explicit methods, called Adams methods, in the absence of stiffness, and a class of implicit methods, the Backward Differentiation Formulae (BDF), otherwise. In order to efficiently switch between these two methods, LSODA initially assumes the problem to be non-stiff and collects data at the end of each integration step; if the problem changes character during the integration step, LSODA switches to the most efficient method. Three functional settings control the performance of LSODA and the quality of the integration, namely, the absolute and relative error tolerance and the maximum number of internal iterations allowed for each integration point.

4.4 Chemical Master Equation

If the biochemical system Ω is characterized by the presence of species appearing in low numbers, their quantities cannot be approximated with concentrations and the impact of noise (*i.e.*, stochastic fluctuations in the number of molecules) becomes significant. In these cases, stochastic approaches are preferred, as they can capture the discrete nature of the system's components and they take into account the influence of stochastic fluctuations on the system's behavior. This feature allows to investigate emergent phenomena that are due to the presence of random noise in the system, such as bistability.

One of these approaches is modeling Ω by means of a Chemical Master Equation (CME) [63], which describes the probability of finding the system in a particular state $\mathbf{X}(t)$ at time t, starting from a given initial state $\mathbf{X}_0 = \mathbf{X}(t_0)$. Formally, the CME is defined as:

$$\frac{\partial \mathbb{P}(\mathbf{X}, t|\mathbf{X}_0, t_0)}{\partial t} = \sum_{j=1}^{p} [a_j(\mathbf{X} - \boldsymbol{\nu}_j)\ \mathbb{P}(\mathbf{X} - \boldsymbol{\nu}_j, t|\mathbf{X}_0, t_0) - a_j(\mathbf{X})\ \mathbb{P}(\mathbf{X}, t|\mathbf{X}_0, t_0)], \quad (4.2)$$

where $a_j(\mathbf{X}(t)) = c_j \cdot d_j(t)$ is the propensity function of reaction P_j at time t, and d_j represents the number of distinct combinations of the reactant molecules appearing

in reaction P_j and occurring in the system at time t.

4.5 Stochastic simulation algorithms

Although the CME allows to compute the probability $\mathbb{P}(\mathbf{X}(t))$ given an arbitrary time t and initial state $\mathbf{X}_0$, its analytical solution is impractical in most cases. As a matter of fact, solving the CME involves computing the probabilities of all the possible states that the system could reach, whose number grows exponentially with the number of chemical species (for some systems, this number might even be infinite [64, 65]). For this reason, analytical and numerical solutions are often intractable, or possible only in the case of simple models, with limited practical applications [66].

An alternative approach is represented by sampling possible trajectories of the underlying Markov-chain process, by means of stochastic simulation algorithms. One of the first method to generate exact realizations of the CME was Gillespie's Stochastic Simulation Algorithm (SSA) [67]. Given a state $\mathbf{X}(t) \in \mathbb{N}^m$ of the biochemical system, SSA determines which reaction will be executed during the next time interval $[t, t+\tau)$ by calculating the probability of each reaction to occur in the next infinitesimal time step $[t, t+dt)$. Such probability is proportional to $a_j(\mathbf{X}(t))dt$, where $a_j(\mathbf{X}(t))$ is the propensity function of reaction P_j at time t. Then, the waiting time τ before a reaction fires is calculated according to the following equation:

$$\tau = \frac{1}{\sum_{j=1}^{p} a_j(\mathbf{X}(t))} \ln\left(\frac{1}{\rho_1}\right), \tag{4.3}$$

while the index j of the firing reaction is chosen as the smallest integer j such that:

$$\sum_{j'=1}^{j-1} a_{j'}(\mathbf{X}(t)) < \rho_2 \left(\sum_{j=1}^{p} a_j(\mathbf{X}(t))\right) \leq \sum_{j'=1}^{j} a_{j'}(\mathbf{X}(t)). \tag{4.4}$$

In Equations 4.3 and 4.4, ρ_1 and ρ_2 are two random numbers sampled from a uniform distribution in the interval $(0, 1]$. Once τ and the index j are chosen using Equations 4.3 and 4.4, the amounts of the species are updated according to reaction P_j: α_{ij} molecules of x_i are removed and β_{ij} molecules of x_i are added to the system state $\mathbf{X}(t+\tau)$, for all $i = 1, \ldots, m$. The simulation time t is then updated to $t+\tau$, and the algorithm is iterated until t exceeds the maximum simulation time t_{max}.

After the original formulation of SSA, the so-called "Direct Method" [67], several variations and improvements of this algorithm were proposed, notably the First Reaction Method [68], the Next Reaction Method [69], and the Tau-leaping algorithm [70]. Other stochastic simulation approaches, not based on SSA, are also exploited in the field of biochemical modeling, such as Stochastic Differential Equations [71], and objects based methods [65, 72].

Part II

Novel work

Chapter 5

Dynamic fuzzy modeling

Dynamic models of complex systems are used in several disciplines to elucidate emergent properties in normal and perturbed conditions, reveal possible failures and/or counter-intuitive mechanisms, and envisage new hypotheses that can be tested. Although mechanism-based models provide a detailed description of the underlying processes, they require a profound knowledge of the system under investigation, including quantitative parameters (*e.g.*, kinetic rates) that are often difficult to be measured, especially in natural or large-scale systems, therefore hampering the effectiveness of many computational analyses. Moreover, some systems might be characterized by the presence of heterogeneous components (*e.g.*, represented by real and non-real valued variables, spanning multiple orders of magnitude, having different units of measures), and/or the data available for them might be present only in a qualitative form, or as a "fuzzy" concept, as described in Chapter 2, highlighting uncertainty and experimental measurement limitations.

The tools provided by fuzzy logic and fuzzy reasoning can overcome these limitations in complex systems modeling [11, 21, 49]. Just to mention a few, models based on fuzzy logic were applied to model cellular signaling pathways [73], stock market [74], or control industrial processes [75, 76]. However, to date a general purpose simulator of dynamic fuzzy models is still missing. This chapter introduces dynamic fuzzy models (DFMs), a computational paradigm to describe and analyze the emergent behavior of heterogeneous complex systems characterized by uncertainty, and a novel general purpose simulator of DFMs. In addition, a description of how to couple such simulator to global optimization techniques, in order to automatically optimize a desired system's behavior, is also provided.

In order to prove the effectiveness of DFMs, an application of this computational method is presented in Section 8.1 for the analysis of programmed cell death in cancer cells, in order to understand the glucose-dependent mechanisms driving cancer cells to death or survival. This system is characterized by the presence of heterogeneous components, for whom precise quantitative information is often missing. This work shows how the coupling of DFM simulation and a global optimization algorithm can guide the design of novel chemotherapies, by combining existing drugs directed to well-established cancer specific targets. The modeling framework introduced in this chapter, together with its application to the study of programmed cell

death, has been extensively presented in [77].

5.1 DFM definition and simulation

5.1.1 Model definition

A DFM of a complex system Ω consists of a set of linguistic variables $\mathcal{L}$ and a set of fuzzy rules $\mathcal{R}$. Each linguistic variable, together with its associated linguistic terms, describes a component of Ω, providing a qualitative description of all the possible states that the component can assume over time. The linguistic variables in $\mathcal{L}$ and their terms are used to define the fuzzy rules in $\mathcal{R}$, which provide a qualitative description of the mechanisms (*e.g.*, feedback regulation) driving the overall behavior of the system.

The fuzzy rules in $\mathcal{R}$ are written as in Expression 2.1, and they are 0-order Sugeno rules, thus their consequent is a constant function. The value of this constant function is defined as "output crisp value". Output crisp values are associated to a unique string of text, which can be used for the definition of fuzzy rules. A 0-order Sugeno rule base has been chosen in order to run computationally efficient simulations of the model, while, at the same time, maintaining a high degree of interpretability of the rules thanks to the use of constant functions associated to output crisp values.

A DFM can be considered as a fuzzy network (FN) [49], and thus depicted as a directed graph, where nodes represent linguistic variables, and arcs the presence of some fuzzy rules governing them. The arcs either represent positive or negative regulations existing among the variables. Consequently, DFMs can be defined with arbitrary topologies, including cycles and feedback loops. The variables belonging to a DFM are partitioned into two sets: *outer* and *inner* variables. The set of outer variables contains *input* and *output* variables, which can only appear as antecedents and consequents of fuzzy rules, respectively; in other words, input variables are represented by nodes without ingoing edges, while output variables by nodes without outgoing edges. In particular, input variables correspond to the components that trigger the dynamic evolution of the system, while output variables represent the components of interest for the analysis of the system (*e.g.*, some experimentally measurable component). On the contrary, inner variables can appear on both sides of fuzzy rules, and they are used to represent mutual regulations among the system components.

Since the input variables appear only in the antecedents, their state is not updated as a result of fuzzy inference. The state of such variables is set by means of a function over time, also called "input function", which describes the evolution of an input variable throughout the simulation. Thus, let $\mathcal{L}_{in} \subset \mathcal{L}$ denote the subset of input variables that do not appear in the consequent of any fuzzy rule belonging to $\mathcal{F}$, and $\mathcal{T} = [t_0, \ldots, t_{max}]$ denote the sequence of time steps of the model simulation,

as described in Subsection 5.1.2. For each $l_n \in \mathcal{L}_{in}$, $\phi_n : \mathcal{T} \to \mathcal{U}_n$ denotes the input function of l_n, where $\mathcal{U}_n$ represents its universe of discourse.

5.1.2 Simulation

Let $\mathcal{Y} = \{y_1, \ldots, y_N\}$ be the set of initial values of each linguistic variable, that is, the numerical value $y_n \in \mathcal{U}_n$ that each linguistic variable $l_n \in \mathcal{L}$ assumes at t_0. The simulation of a DFM is performed in discrete time steps, starting from t_0 and updating the values of each $l_n \in \mathcal{L}$ synchronously.

Namely, given a time step $t \in \mathcal{T}$, the state of the DFM in t is defined as $S(t) = S^t = [s_1^t, \ldots, s_N^t]$, where $s_n^t = s_n(t)$, $s_n^t \in \mathcal{U}_n$, is the numerical value of l_n in t. The transition $S^t \to S^{t+1}$ is performed by updating the values s_n^{t+1} of each linguistic variable l_n as follows:

- if $l_n \in \mathcal{L}_{in}$, then $s_n^{t+1} = \phi_n(t+1)$;
- otherwise, s_n^{t+1} is obtained by applying the Sugeno fuzzy inference method as described in Section 2.4.2.

The process iterates until the maximum simulation time t_{max} is reached.

To explore the emergent behavior of the system in different scenarios, FUMOSO allows to define "perturbations" of the DFM over one or more time intervals. For the sake of conciseness, in what follows only perturbations over a single time interval are considered.

Formally, to define a perturbation π, consider a subset of perturbable linguistic variables $\widetilde{\mathcal{L}} \subseteq \mathcal{L}$, with $|\widetilde{\mathcal{L}}| = \widetilde{N}$, and a chosen time interval $[t_b, t_e] \subseteq [t_0, t_{max}]$. A perturbation is defined as the set of couples $\pi = \{(\widetilde{l}_n, \widetilde{\phi}_n) | 1 \leq n \leq \widetilde{N}\}$, where $\widetilde{l}_n \in \widetilde{\mathcal{L}}$ and $\widetilde{\phi}_n$ is a user-defined function that determines the value of $\widetilde{l}_n$ over $[t_b, t_e]$. Although perturbations may be defined by means of arbitrary functions in FUMOSO, for the remaining part of this thesis the function $\widetilde{\phi}_n$ associated to $\widetilde{l}_n$ will be equal either to any element of a given finite set of states, composed of the crisp values (*i.e.*, constant functions) associated to $\widetilde{l}_n$, or to the term "unperturbed", a state indicating that the perturbable variable will be actually updated by using the Sugeno inference. This definition of perturbation, which includes the "unperturbed" state, will prove useful to carry out the global optimization of the DFM and for the identification of a (potentially) minimal set of variables to be perturbed.

5.2 FUMOSO: a general-purpose simulator of DFMs

The FUzzy MOdels SimulatOr (FUMOSO) is a novel, open source and cross-platform software, specifically designed to define, simulate and analyse DFMs. FUMOSO is provided with an intuitive Graphical User Interface (GUI), devised in order to guide the user through the required steps for the creation of a DFM, that is, the definition of linguistic variables, together with their linguistic terms and membership functions,

output crisp values, and fuzzy rules (see Figure 5.1). Additionally, FUMOSO allows the user to enter all the information required to simulate the dynamics of a DFM:

1. the simulation interval $[t_0, t_{max}]$;
2. the initial state of all variables;
3. the functions that drive the dynamics of input variables (if any);
4. the perturbation functions (if any).

Optionally, for each variable involved in a perturbation π, the user can specify a list of time intervals in which that perturbation becomes active over that variable.

Once the simulation has ended, FUMOSO plots the dynamics of any chosen system component. Moreover, FUMOSO allows to plot its membership functions and the degree of satisfaction of the firing rules, involved in the update of a component state at any arbitrary time step. The analysis of the simulation outcome is facilitated thanks to the possibility of creating groups of components, that is, subsets of variables whose dynamics are shown in the same plot.

FUMOSO was implemented using the Python programming language [78], version 2.7. FUMOSO's dependencies are: PyQT4, numpy [79], and PyFuzzy [80]. The latter module—used to handle the membership functions evaluations—requires the Python runtime for the ANTLR 3 language processing library [81]. In order to reduce the dependencies and simplify the installation, a porting to the Simpful fuzzy library is under development [82] (see Chapter 6).

FUMOSO currently supports the import and export of fuzzy models encoded using the Fuzzy Control Language (FCL), standard IEC 61131-7 [83]. Since the FCL file does not encode some of the information required to implement a DFM (notably, the initial states of variables, the input functions and the perturbations), this additional data is saved into an external project file with extension .fms. FUMOSO provides an Application Programming Interface (API) to define new models, load models previously saved as .fms files, run simulations in unperturbed or perturbed conditions, and perform a global optimization analysis (see Section 5.3). The support for the Fuzzy Markup Language (FML, IEEE standard 1855-2016 [84]) for models storage is currently under development. The source code of FUMOSO is available under the GPL 2.0 license on GitHub at the following URL: `https://github.com/aresio/FUMOSO`.

5.3 Global optimization of DFMs

Given a DFM, FUMOSO allows to automatically explore the emergent behavior of the system in different scenarios, where the state of one or more linguistic variables is altered in order to simulate the effects of perturbations on some desired system behaviors.

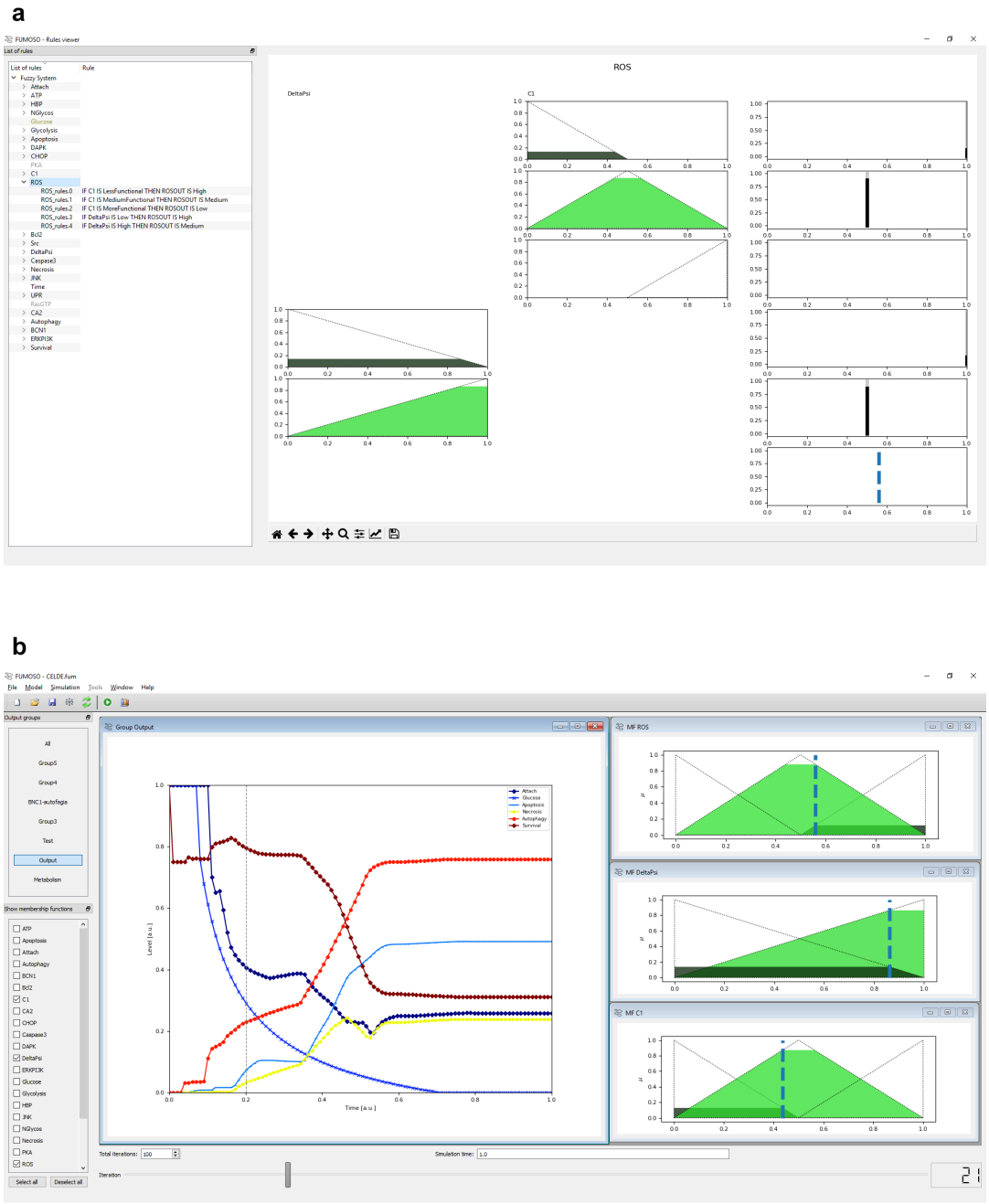


FIGURE 5.1: Graphical User Interface of FUMOSO. (a) Fuzzy rules visualization. The list of fuzzy rules is shown in a dockable subwindow on the left. Rules are grouped according to the consequent variable. By clicking on a set of rules (*e.g.*, "ROS" in the figure), FUMOSO shows the membership functions of all antecedent variables, along with the values of the crisp consequents (on the right). The last row shows the final value calculated by the Sugeno inference method. (b) FUMOSO main window. Besides the menu on the top—which allows to create, open, edit, and save DFMs—the Graphical User Interface consists in three sections: a subwindow listing any user-defined group of model components, a subwindow listing the membership functions defined in the current DFM, the plots for selected groups and selected membership functions. All subwindows are dockable.

In general, a user-defined function $F(\pi)$ can be employed to evaluate the effectiveness of π in obtaining the desired system behavior. However, the search space

of possible perturbations of a DFM grows exponentially with the number of perturbable variables. For example, in a DFM with $\widetilde{N}$ perturbable variables, each characterized by $k+1$ possible perturbed states (k crisp values, plus the term "unperturbed"), the possible perturbations are $(k+1)^{\widetilde{N}}$. Thus, when dealing with DFMs characterized by a high number of variables and/or possible perturbed states, performing an exhaustive search might not be feasible and the use of optimization methods relying on local search strategies might return unreliable results. To face these challenges, global optimization methods can be adopted to realize an effective and efficient exploration of such huge search space. In particular, in this thesis FUMOSO has been coupled with SA [56] (Section 3.2), a stochastic optimization algorithm suitable for discrete optimization problems, as in the case of the optimization of a desired DFM behavior, since the set of possible perturbations is finite.

FUMOSO employs $F(\pi)$ as the objective function of SA. The SA procedure integrated in FUMOSO starts from an initial perturbation π_0—where all the variables in $\widetilde{\mathcal{L}}$ are set to the "unperturbed" state—and then explores the neighborhood of such perturbation in the search space. During the i-th iteration of SA, a new putative perturbation $\pi_i' = \eta_e(\pi_i)$ is generated, where η_e is a neighborhood function that randomly modifies the current perturbation by changing the perturbed state of at most e variables belonging to $\widetilde{\mathcal{L}}$. To be more precise, the perturbation π' is obtained by randomly selecting $e \leq \widetilde{N}$ variables, and setting their state to one of the possible perturbed states (*i.e.*, their output crisp values or "unperturbed"), chosen at random. The "unperturbed" state is leveraged by SA during the optimization to avoid the generation of solutions with an excessive number of perturbations, which might be not strictly necessary to obtain the desired system behavior. So doing, and thanks to the use of the "unperturbed" state, SA can generate the minimal set of variables that need to be effectively perturbed in order to maximize or minimize the given objective function. The output produced by a SA run is the perturbation π characterized by the best value of $F(\pi)$.

Chapter 6

Simpful

The huge impact that fuzzy set theory and fuzzy logic had in several scientific fields, especially related to computer science and engineering [21], brought to the development of several methods, frameworks and software tools involving fuzzy sets or full FISs, usually aimed at specific applications. On the contrary, examples of general software libraries and toolboxes to handle fuzzy sets and/or fuzzy logic have been limited in number and scope. The reasons for this shortcoming can be found in the difficulties of dealing with the complex objects required by fuzzy logic (*i.e.*, fuzzy sets, fuzzy rules and natural language), and in the wide number of existing types of FISs and their applications.

Among the existing softwares, one of the most popular is the Fuzzy Logic Toolbox in Matlab [85]. This toolbox offers a comprehensive and up to date list of functions for managing a wide variety of applications involving fuzzy logic, but it is commercially distributed only. Thus, in the last years open source alternatives were developed by the scientific community.

Although numerous, most of these alternatives are outdated, discontinued or designed for specific applications. PyFuzzy [80] was the first general purpose library for designing FISs, developed using the Python programming language (version 2.7). This library allows to manage all the entities needed to construct FISs, and supports the creation of numerous types of fuzzy sets and FISs. Moreover, PyFuzzy supports the export and sharing of FISs by means of FCL files. FCL files implement the old standard IEC 61131 (IEC61131-7) [83], which was designed for fuzzy control applications and remained for many years the only *de facto* standard for representing FISs. Despite its completeness, PyFuzzy is now outdated and not maintained anymore.

Most of the recent open-source softwares for fuzzy logic are aimed at data driven applications, including machine learning, classification and regression analysis, or decision support systems. Among the most popular, one can find: FuzzyLite [86], a collection of C++ libraries designed for fuzzy control; Fispro [87], a C++ software provided with a GUI, specifically designed for data driven applications of FISs and their automatic learning from a dataset; Juzzy [88], a Java based toolkit for handling type-2 fuzzy sets. One of the most recent and interesting implementations in the

fuzzy community is represented by the JFML library [89], the only open source library (implemented in Java) incorporating the most recently developed standard for representing FISs, IEEE 1855-2016 [90], which defines a new W3C eXtensible Markup Language (XML)-based language, named FML [91].

Excluding the outdated and discountinued PyFuzzy, a general purpose, open source, and intuitive Python library is still missing. To overcome these limitations, during this thesis project, a new general purpose library for fuzzy logic, named Simpful, was developed. Simpful is a Python library for fuzzy reasoning, designed to provide a simple and lightweight Application Programming Interface (API), as close as possible to natural language. Simpful allows to define FISs for any purpose, providing a set of classes and methods to intuitively define and handle fuzzy sets, fuzzy rules and perform fuzzy inference. In particular, fuzzy rules can be defined by means of strings of text in natural language, simplifying the definition of fuzzy rule bases.

Although not limited to it, Simpful allows to define and simulate DFMs of complex systems. This is achieved by supporting the creation of FNs, in which some FIS output can be fed as input to a different FIS (see Section 2.5), and supporting fuzzy inference across multiple components of the network. These networks can be characterized by any arbitrary topologies, including cycles and feedback loops, a feature that proves useful to represent the interactions existing in complex systems (for a few examples, see Section 6.2, or Chapters 5 and 7). A detailed description of this novel library and some practical examples will appear in [82].

6.1 Implementation and supported features

Simpful was implemented using the Python programming language, and it is compatible with both versions 2.7 and 3.7 [78]; its dependencies are numpy [79] and scipy [92]. The latest version of Simpful currently supports the following features:

- the definition of polygonal fuzzy sets;
- the definition of fuzzy rules as strings of text written in natural language;
- the logic operators AND, OR, NOT;
- the Sugeno inference method.

A fuzzy set is defined as an ordered list of points in a plane, where the first coordinate is their value in the universe of discourse, while the second represents the degree of membership, ultimately identifying a polygon. Simpful performs an interpolation between these points to identify the membership function characterizing the fuzzy set, and possibly allowing for fuzzy sets with infinite support over their universe of discourse. The user is also required to associate to each fuzzy set a meaningful linguistic term. The defined fuzzy sets can be added to a linguistic variable object, provided with its own name, as given by the user.

The user has to define also the functions exploited by the Sugeno inference. In the case of 0-order functions, *i.e.*, constant functions, these are defined as "output crisp values". In the case of higher order Sugeno systems, the user can define "output functions" as strings of text: these will be evaluated by exploiting the `eval()` function of Python, which parses the expression given as a string argument and executes it as code within the program. Both output crisp values and output functions must have an associated meaningful and unique string to identify them, which will be exploited in the definition of the fuzzy rules.

Fuzzy rules are defined by well formed strings, written in natural language and in the form showed in Expression 2.1. The logic operators are currently defined as in Section 2.3. The rules must contain the name associated to the linguistic variable objects, use linguistic terms contained in them, and the strings identifying the output crisp values or the output functions defined by the user.

Linguistic variables, fuzzy rules, output crisp values and output functions are added to a fuzzy reasoner object, which implements the whole FIS. Given input values for the variables appearing in the antecedents of the fuzzy rules, the method implementing the Sugeno inference can be called for one or more of the variables appearing in the consequent of fuzzy rules, in order to obtain their final output. The state of the variables and the result of the inference are returned to the user as key-value couples inside a dictionary, where keys represent the names of the variables. Figure 6.1 shows an overview of how to construct a fuzzy reasoner object and how to perform inference in Simpful. The source code of Simpful is available, under GPL license, on GitHub at the following URL: `https://github.com/aresio/simpful`.

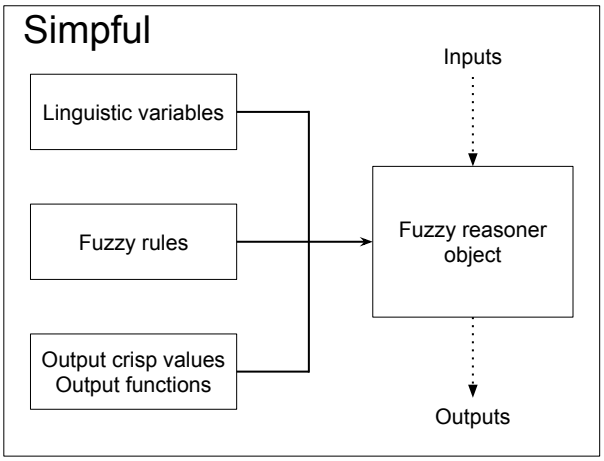


FIGURE 6.1: Graphical representation of the fuzzy reasoner object in Simpful.

Simpful will be extended to support other commonly used types of fuzzy sets, including non-polygonal fuzzy sets, such as Gaussian and sigmoidal ones. Moreover, the support for additional types of fuzzy inference methods is also planned, including for example the Mamdani [43], the Tsukamoto [93] and the AnYa [94] methods. Finally, in the future Simpful will implement the FML format defined in the IEEE

Std 1855-2016 [90], in order to facilitate the import, export and sharing of the FISs defined within this library.

6.2 Examples

In what follows, some example code of Simpful is provided.

6.2.1 Tipping problem

The tipping problem consists in computing the appropriate tip in a restaurant, taking into account as inputs the food and service quality.

Listing 6.1 shows how to define a simple FIS for the tipping problem by means of Simpful. In line 5 a fuzzy reasoner object is created. The fuzzy sets and the linguistic variable "Service" are defined in lines 8 to 11; this variable contains three fuzzy sets, "poor", "good" and "excellent", ranging from 0 to 10. The universe of discourse is not explicitly defined. In the current implementation, fuzzy sets are supposed to have infinite support, thus an input value outside the defined polygonal fuzzy set will be fuzzified as its closest point to the polygonal fuzzy set (*e.g.*, $\mu_{excellent}(11) = 1$). From line 13 to 15 the linguistic variable for food quality is defined, exploiting two fuzzy sets, "rancid" and "delicious". Fuzzy output crisp values "cheap", "average" and the output function "generous", referring to the tip amount, are defined in lines 18 to 20. Fuzzy rules are defined in lines 23 to 26. Once the input values are set (in this example, lines 29 and 30, "Service" and "Food" quality scored 4 and 8 points, respectively), fuzzy inference is performed in line 33 to obtain the final tipping amount, which is equal to 14.77% in this example.

LISTING 6.1: A FIS for the tipping problem, defined in Simpful

```
1  from simpful import *
2
3  # A simple fuzzy inference system for the tipping problem
4  # Create a fuzzy reasoner object
5  FR = FuzzyReasoner()
6
7  # Define fuzzy sets and linguistic variables
8  S_1 = FuzzySet( points=[[0., 1.], [5., 0.]], term="poor" )
9  S_2 = FuzzySet( points=[[0., 0.], [5., 1.], [10., 0.]], term="good" )
10 S_3 = FuzzySet( points=[[5., 0.], [10., 1.]], term="excellent" )
11 FR.add_membership_function("Service", MembershipFunction( [S_1, S_2, S_3],
       concept="Service quality" ))
12
13 F_1 = FuzzySet( points=[[0., 1.], [10., 0.]], term="rancid" )
14 F_2 = FuzzySet( points=[[0., 0.], [10., 1.]], term="delicoius" )
15 FR.add_membership_function("Food", MembershipFunction( [F_1, F_2],
       concept="Food quality" ))
16
17 # Define output crisp values
18 FR.set_crisp_output_value("cheap", 5)
19 FR.set_crisp_output_value("average", 15)
20 FR.set_output_function("generous", "Food+Service+5")
21
```

```
22 # Define fuzzy rules
23 R1 = "IF ((Service IS poor) OR (Food IS rancid)) THEN (Tip IS cheap)"
24 R2 = "IF (Service IS good) THEN (Tip IS average)"
25 R3 = "IF ((Service IS excellent) OR (Food IS delicoius)) THEN (Tip IS
       generous)"
26 FR.add_rules([R1, R2, R3])
27
28 # Set antecedents values
29 FR.set_variable("Service", 4)
30 FR.set_variable("Food", 8)
31
32 # Perform Sugeno inference and print output values
33 print FR.Sugeno_inference(['Tip'])
```

6.2.2 Repressilator

The repressilator is a synthetic genetic regulatory network consisting in three genes placed in a feedback loop, where the genetic product of each gene inhibits the expression of the next gene in the network (the system is depicted in Figure 6.2). The repressilator was designed, studied by means of mechanistic modeling, and finally implemented *in vivo* by exploiting a strain of the bacterium *E. coli* for the first time in [95]. This simple system was specifically designed to exhibit a stable oscillatory regime, thanks to the presence of the negative feedback loop, a behavior that was successfully reproduced both in the simulations of the model and in the *in vivo* experiments. Here, a simple redefinition of the original model in terms of a DFM is provided, in order to show how Simpful can be applied to this kind of fuzzy modeling of complex systems.

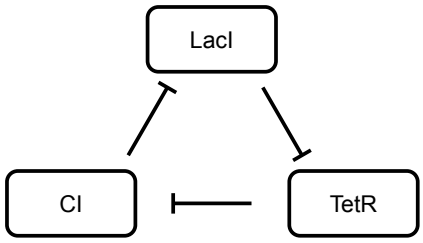


FIGURE 6.2: Graphical representation of the repressilator. Blunt arrows represent inhibitory interactions.

The example code of the repressilator is given in Listing 6.1. In line 6 a fuzzy reasoner object is created. From line 9 to 19, the three linguistic variables related to the three species constituting the repressilator are defined. In particular, all three species are characterized by a universe of discourse ranging from 0 to 1, and by the presence of two fuzzy sets, "low" and "high", representing the quantity of each protein. In this example, a value of 1 in the universe of discourse corresponds to the maximum quantity, while 0 to the absence of the protein. Analogously, the output crisp values are defined in lines 23 and 24, by setting "low" to 0 and "high" to 1. Lines 27 to 34 contain the definition of the fuzzy rules, representing the negative feedbacks existing between the three proteins.

LISTING 6.2: A DFM of the repressilator, defined in Simpful

```
1  from simpful import *
2  from copy import deepcopy
3
4  # A simple dynamic fuzzy model of the repressilator
5  # Create a fuzzy reasoner object
6  FR = FuzzyReasoner()
7
8  # Define fuzzy sets and linguistic variables
9  L_1 = FuzzySet( points=[[0., 1.], [1., 0.]], term="low" )
10 L_2 = FuzzySet( points=[[0., 0.], [1., 1.]], term="high" )
11 FR.add_membership_function("LacI", MembershipFunction( [L_1, L_2],
       concept="LacI quantity" ))
12
13 T_1 = FuzzySet( points=[[0., 1.], [1., 0.]], term="low" )
14 T_2 = FuzzySet( points=[[0., 0.], [1., 1.]], term="high" )
15 FR.add_membership_function("TetR", MembershipFunction( [T_1, T_2],
       concept="TetR quantity" ))
16
17 C_1 = FuzzySet( points=[[0., 1.], [1., 0.]], term="low" )
18 C_2 = FuzzySet( points=[[0., 0.], [1., 1.]], term="high" )
19 FR.add_membership_function("CI", MembershipFunction( [C_1, C_2],
       concept="CI quantity" ))
20
21
22 # Define output crisp values
23 FR.set_crisp_output_value("low", 0.0)
24 FR.set_crisp_output_value("high", 1.0)
25
26 # Define fuzzy rules
27 RULES = []
28 RULES.append("IF (LacI IS low) THEN (TetR IS high)")
29 RULES.append("IF (LacI IS high) THEN (TetR IS low)")
30 RULES.append("IF (TetR IS low) THEN (CI IS high)")
31 RULES.append("IF (TetR IS high) THEN (CI IS low)")
32 RULES.append("IF (CI IS low) THEN (LacI IS high)")
33 RULES.append("IF (CI IS high) THEN (LacI IS low)")
34 FR.add_rules(RULES)
35
36 # Set antecedents values
37 FR.set_variable("LacI", 1.0)
38 FR.set_variable("TetR", 0.5)
39 FR.set_variable("CI", 0.0)
40
41 # Set simulation steps and save initial state
42 steps = 14
43 dynamics = []
44 dynamics.append(deepcopy(FR._variables))
45
46 # At each simulation step, perform Sugeno inference, update state and save
       the results
47 for i in range(steps):
48         new_values  = FR.Sugeno_inference()
49         FR._variables.update(new_values)
50         dynamics.append(new_values)
```

The initial state of the DFM is set in lines 37 to 39, the number of simulation steps is defined in line 42, while the data structure containing the results of the simulation is initialized in lines 43 and 44. Lines 47 to 50 contain the `for` loop in which the

simulation is performed. In particular, the new state of the system is inferred in line 48, updated in line 49 and then stored in the previously defined data structure. The final output of the simulation can be plotted as shown in Figure 6.3. Despite its simplicity and the lack of a precise kinetic parameterization, this model can reproduce the typical oscillatory dynamics of the three species already shown in the original model.

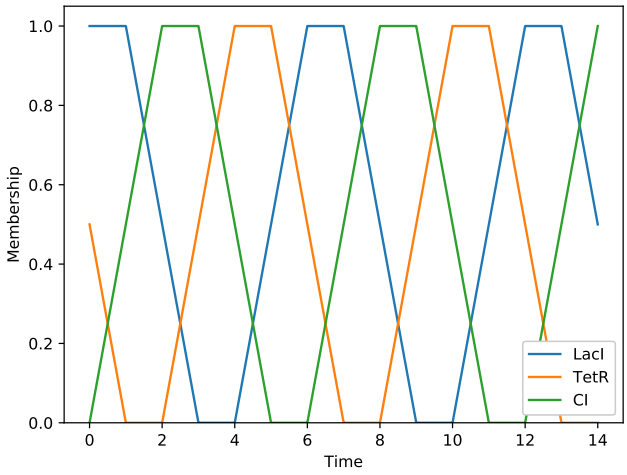


FIGURE 6.3: Dynamics of the species of the DFM representing the repressilator.

Chapter 7

Hybrid modeling with FuzzX

In the modeling and simulation of complex systems, a common problem faced by modelers is that the knowledge and the data regarding the system's components and interactions might be available in various types and forms, hindering their integration inside a unified modeling framework. When dealing with such a variety of information, different mathematical formalisms could be potentially exploited and connected together to represent different parts of the modeled system. This is particularly relevant when mechanistic information is available for some parts of the system, since mechanistic models are closer to the physical reality of the modeled phenomena and generally introduce less approximations both on the system description and the outcome of the computational analysis.

In this context, the advantages provided by fuzzy logic, and in particular the flexibility of fuzzy sets, can prove very useful for bridging together different types of information. Indeed, several approaches made use of fuzzy logic to tackle this problem, with different strategies. A few examples include the approximation of complex non-linear functions with fuzzy logic [96] or Adaptive-Network-Based Fuzzy Inference System [97], the fuzzy aggregation of a set of local linear models [98], the modeling of complex systems with Fuzzy Cognitive Maps that oversee and control different local models [99], and the definition of fuzzy lattices [100], which exploit lattice computing [101] to define decision making systems that take into account heterogeneous type of data. Fuzzy logic was also exploited to approximate missing parameters [102] in existing mechanistic approaches (see, *e.g.*, Fuzzy Petri Nets [103]).

These approaches employ fuzzy logic to improve existing formalisms and address their limitations, or aggregate the results of different local models. However, none of them deals with the definition of different sub-models that represent both qualitative and mechanistic knowledge, and that are able to generate the emergent system dynamics by an iterative exchange of information. Attempts to bring together mechanistic modeling with fuzzy systems within a unified framework have been limited so far (an example of ODE models coupled to fuzzy inference systems can be found in [104]). Even when considering approaches that do not involve fuzzy logic, attempts to reconcile qualitative and quantitative models were made by defining new formalisms (such as the ANIMO framework [105]), or by automatically transforming logic-based models in ODEs by means of piece-wise differential

equations [106] or multivariate polynomial interpolation [107].

Differently from the approaches mentioned above, the framework presented in this chapter, named Fuzzy-mechanistic modeling of compleX systems (FuzzX), represents the first attempt to bring together mechanistic modeling with fuzzy systems in a unique, general-purpose and dynamic framework in which two different regimes (*i.e.*, mechanistic and fuzzy) control each other. Specifically, FuzzX is a novel computational framework for the definition and simulation of hybrid models of complex dynamical systems, composed of a mechanistic module and a fuzzy module. The mechanistic module can be defined by means of any kind of fully parameterized modeling formalism (*e.g.*, algebraic equations, ODEs, Markov jump processes). Any variable of the mechanistic module can serve as input for the qualitative module, which is formalized as a FN [49] and is in turn able to control either some variables or parameters of the mechanistic module. The key peculiarity of FuzzX is that it allows to define, within the hybrid model, feedback regulations that simultaneously involve both the mechanistic and the fuzzy regimes. FuzzX provides modelers with an effective methodology to couple the description and analysis of well-known and detailed processes, along with other phenomena whose functioning is not well characterized and can only be described by means of linguistic concepts. FuzzX is a completely general-purpose approach that can be applied to *any* complex system. In the next sections, a general description and notation of this novel formalism is provided.

In order to show its potentiality, FuzzX has been applied to redefine, in terms of a hybrid model, a stochastic mechanistic model of a biochemical signaling pathway that is characterized by complex non-linear behaviors arising from several feedback regulations among the system components [108]. The model definition, its simulation and its analysis are reported in Section 8.2. This application shows that, albeit mechanistic interactions are substituted by a set of expert-defined fuzzy rules, the hybrid model is able to reproduce the emergent behaviors of the system, such as the presence of a transient phase and the establishment of stable or damped oscillations, both in normal and perturbed conditions.

A complete description of FuzzX and its application can be found in [109, 110].

7.1 Hybrid models definition and simulation

7.1.1 Model definition

In FuzzX, a hybrid model of a complex system Ω is defined by specifying two components: a *mechanistic module* $\mathcal{M}$ and a *fuzzy module* $\mathcal{F}$. The module $\mathcal{M} = \langle \mathcal{V}^{\mathtt{M}}, \theta^{\mathtt{M}}, \mathcal{P} \rangle$ consists in three disjoint sets corresponding to, respectively, the set of variables, the set of parameters and the set of mechanistic processes governing the functioning of Ω. $\mathcal{M}$ is used to model all those processes of the system that are known at a high level of detail. The module $\mathcal{F} = \langle \mathcal{V}^{\mathtt{F}}, \mathcal{R} \rangle$ consists in two disjoint sets corresponding

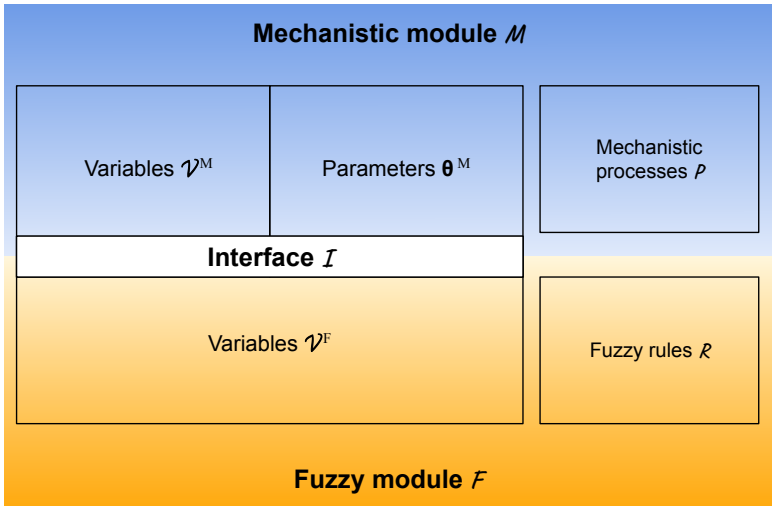


FIGURE 7.1: Scheme of the FuzzX framework for hybrid modeling. The mechanistic and fuzzy modules are represented in blue and orange color, respectively.

to, respectively, the set of linguistic variables and the set of fuzzy rules. $\mathcal{F}$ is used to represent all the processes that are not known in full detail, cannot be represented by crisp quantities, or require a higher degree of approximation. Importantly, in FuzzX the *interface* of Ω is defined as the set $\mathcal{I} = (\mathcal{V}^{\mathrm{M}} \cap \mathcal{V}^{\mathrm{F}}) \cup (\theta^{\mathrm{M}} \cap \mathcal{V}^{\mathrm{F}})$. The interface between the mechanistic and the fuzzy modules allows them to communicate, effectively resulting in the two modules influencing the dynamic behavior of each other, as explained in what follows. It should be noted that the module $\mathcal{M}$ treats elements appearing in the interface either as variables (if they belong to the set $\mathcal{V}^{\mathrm{M}} \cap \mathcal{V}^{\mathrm{F}}$) or parameters (if they belong to the set $\theta^{\mathrm{M}} \cap \mathcal{V}^{\mathrm{F}}$), while the module $\mathcal{F}$ treats all the elements of the interface as fuzzy variables. This is indeed a peculiar feature of this modeling framework: since both *parameters* and *variables* of the mechanistic module can be treated as fuzzy variables by the fuzzy module, FuzzX can be employed for the modeling of the dynamic behavior of complex systems that requires handling both precise and uncertain data. Figure 7.1 shows a graphical schematization of the two modules and their interface, as defined in FuzzX.

Let us denote by $\mathcal{V}^{\mathrm{M}} = \{x_1^{\mathrm{M}}, \ldots, x_m^{\mathrm{M}}\}$ and $\mathcal{V}^{\mathrm{F}} = \{x_1^{\mathrm{F}}, \ldots, x_f^{\mathrm{F}}\}$ the elements in the sets of mechanistic and fuzzy variables, respectively. The variables in $\mathcal{V}^{\mathrm{M}}$ can assume values in some given set $\mathcal{X}_M$, according to the mathematical formalism used to define the module $\mathcal{M}$, while the variables in $\mathcal{V}^{\mathrm{F}}$ assume values in $\mathbb{R}$. Then, let $\mathbf{X}^{\mathrm{M}}(t) = (x_1^{\mathrm{M}}(t), \ldots, x_m^{\mathrm{M}}(t))$ denote the state of the mechanistic module at time t, and $\mathbf{X}^{\mathrm{F}}(t) = (x_1^{\mathrm{F}}(t), \ldots, x_f^{\mathrm{F}}(t))$ denote the state of the fuzzy module at time t. Overall, these two vectors represent the state of the system Ω, described as a hybrid model.

7.1.2 Simulation

The update of the states of the mechanistic and fuzzy modules is achieved by means of two functions. The first function, denoted by $U^{\mathrm{M}} : \mathcal{X}_M^m \to \mathcal{X}_M^m$, maps the current

state $\mathbf{X}^{\mathtt{M}}(t)$ of $\mathcal{M}$ into the state $\mathbf{X}^{\mathtt{M}}(t+1)$ at the next time step, by taking into account the parameters in $\theta^{\mathtt{M}}$ and the mechanistic processes expressed in $\mathcal{P}$. Stated otherwise, $U^{\mathtt{M}}$ describes the temporal evolution of the processes described in $\mathcal{M}$ in mechanistic detail. This function can be formally evaluated by means of any computational method that is suitable to the specific mathematical formalism that has been used to define $\mathcal{M}$ (*e.g.*, $U^{\mathtt{M}}$ can be a numerical integration algorithm if $\mathcal{M}$ is formalized as a system of ODEs). The second function, denoted by $U^{\mathtt{F}} : \mathbb{R}^f \to \mathbb{R}^f$, maps the current state $\mathbf{X}^{\mathtt{F}}(t)$ of $\mathcal{F}$ into the state $\mathbf{X}^{\mathtt{F}}(t+1)$ at the next time step, by taking into account the fuzzy rules specified in $\mathcal{R}$. Hence, $U^{\mathtt{F}}$ represents the evolution of parts of the system that are known at a lower level of detail and/or precision.

In FuzzX, a simulation step of the hybrid model of Ω consists in the application of $U^{\mathtt{M}}$ to calculate the dynamics of the mechanistic module, followed by the application of $U^{\mathtt{F}}$ to perform a fuzzy inference (see Figure 7.2). To be more precise, the temporal evolution of $\mathcal{M}$ is simulated by applying the function $U^{\mathtt{M}}$ for a user-defined time interval of length Δ, for some $\Delta \in \mathbb{R}^+$. The choice of Δ determines how frequently the two modules will interact with each other.

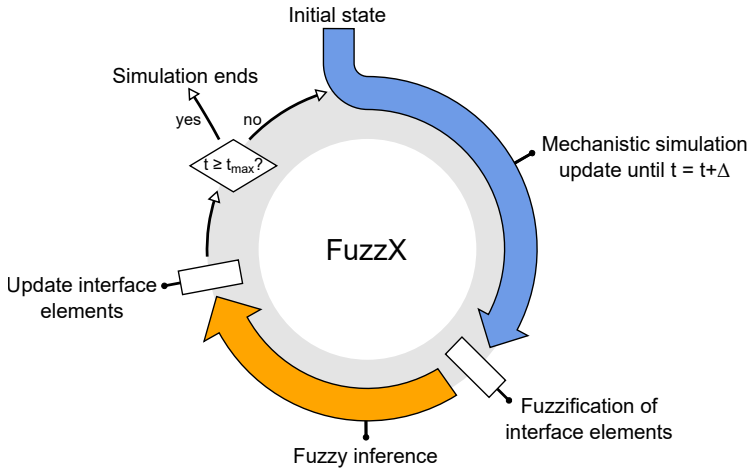


FIGURE 7.2: General procedure for the simulation of a hybrid model defined in FuzzX. The state update of the mechanistic and fuzzy modules are represented in blue and orange color, respectively. White boxes represent the update steps of interface elements, occurring between mechanistic simulation and fuzzy inference, or vice versa.

The application of the functions $U^{\mathtt{M}}$ and $U^{\mathtt{F}}$ on the elements that belong to the interface enables the communication between the two modules, and their mutual regulation. Namely, the value of an interface element $I \in \mathcal{V}^{\mathtt{M}} \subseteq \mathcal{I}$ (*i.e.*, a mechanistic variable $x_1, \ldots, x_m$) is first updated by $U^{\mathtt{M}}$ and then fuzzified. So doing, the state of the fuzzified mechanistic variable at the interface affects the dynamics of the fuzzy module, by means of the fuzzy inference operated by $U^{\mathtt{F}}$. As a matter of fact, the values of any element $I \in \mathcal{I}$ can be updated by $U^{\mathtt{F}}$, thus modifying the values of variables and parameters belonging to $\mathcal{M}$, eventually affecting the dynamics of the mechanistic module during the next step of the simulation.

The simulation of the hybrid model ends when a user-defined time limit $t_{max} \in \mathbb{R}^+$ is reached, which corresponds exactly to $\kappa = \lceil \frac{t_{max}}{\Delta} \rceil$ simulation steps. Without loss of generality, FuzzX can be adapted to simulate hybrid models using time steps of arbitrary length, instead of adopting a uniform advancement of length Δ: in such a case, the function $U^{\mathtt{M}}$ has to be modified appropriately in order to take into account a vector $\mathbf{\Delta} = (\Delta_1, \ldots, \Delta_D)$ of different time intervals.

To sum up, a hybrid model can be defined by using the following workflow:

- *Step 1*: identify all the components of the system and their interactions;
- *Step 2*: assign each component to either the mechanistic or the fuzzy module, and identify the components that will belong to the interface of the hybrid model;
- *Step 3*: describe the variables and the processes governing the mechanistic module with a suitable mathematical formalism (*e.g.*, ODEs, Markov chains, etc.). The description of these mechanistic processes (*e.g.*, interactions among the system components) has to be curated by the user, according to the available knowledge about the system;
- *Step 4*: define the linguistic variables (together with their fuzzy sets and linguistic terms) and the fuzzy rules belonging to the fuzzy module. The universe of discourse of the linguistic variables belonging to the interface must be suitable for, and in accordance with, the formalism employed for the definition of the mechanistic module;
- *Step 5*: choose a suitable value of Δ to control how frequently the two modules will exchange information.

7.2 Implementation details

FuzzX was implemented exploiting the Python programming language, v.2.7 [78]. Its dependencies are numpy [79] and Simpful (see Chapter 6). A porting to Python version 3.7 is currently under development.

In order to properly define a hybrid model and to run simulations, the user must specify a mechanistic and a fuzzy module by means of the provided classes. The mechanistic module is defined by means of the `mech_module` class: its initialization requires a dictionary containing the initial state of the variables, a dictionary containing the parameters of the module and a function that implements the mechanistic simulation; additional arguments required by this latter function can be passed and stored in the `mech_module` object. The fuzzy module is defined as a fuzzy reasoner object in Simpful, as shown in Chapter 6. The main class implementing the hybrid simulation is the `FuzzX` class. Once a `FuzzX` object is created, the mechanistic and fuzzy modules are added to it by means of the `add_mech_module()` and

`add_fuzz_module()` methods, respectively. Finally, a simulation can be performed be means of the `simulate()` method, providing t_{max} and Δ as arguments.

Chapter 8

Applications

In this chapter, two real world applications of the modeling frameworks defined in this thesis are shown. In particular, a DFM of programmed cell death in cancer cells is presented in Section 8.1 (see also [77]), whereas Section 8.2 describes a hybrid model of a biochemical signaling pathway in yeast defined with FuzzX (see also [109]). In addition, two novel applications of DFMs and FuzzX started to be developed in the context of this thesis. A brief description of them is given in Section 8.3.

8.1 A dynamic fuzzy model of programmed cell death

To show the potentiality of DFMs in investigating complex, heterogeneous systems, this section presents a DFM that describes the programmed cell death processes taking place inside oncogenic K-ras cancer cells—characterized by the so-called "Warburg effect"—grown in a progressive limiting amount of glucose. The aim of this model is to understand the glucose-dependent mechanisms driving cancer cells either to death or survival. Briefly, the Warburg effect, or aerobic glycolysis, is a metabolic hallmark of cell malignancy, and it refers to the ability of tumor cells to favor metabolism via glycolysis rather than the much more efficient oxidative phosphorylation pathway (employed in physiological conditions) [111]. Accordingly, numerous cancer cells, grown either in low glucose availability or in free glucose, are strongly susceptible to cell death with respect to normal cells. However, it has also been observed that not all cancer cells undergo cell death when grown in a harsh environment, such as in glucose starvation, since some of them might acquire the ability to survive in this new environmental condition by activating compensatory signaling pathways [112, 113] and alternative metabolic routes [114, 115]. These metabolically rewired cancer cells are often more aggressive [116], and can be selected by chemotherapy, by therapies exploiting synergism between chemotherapeutic treatments and anti-metabolic drugs, or by genetic and pharmacological ablation of oncogenic pathways [117–119]. Overall, the emerging cell behavior is influenced by highly interconnected processes with multiple levels of regulations (*e.g.*, protein-protein interactions and modifications, positive and negative feedbacks), and able to promote opposite effects on cancer cells (*i.e.*, sensitivity vs. resistance to chemotherapy). In this scenario, the combination of therapies targeting aerobic

glycolysis, adaptive mechanisms (*i.e.*, increased autophagy) and well-established cancer specific targets (*i.e.*, tyrosine kinase signaling pathways) represent a potential approach to be explored in cancer cure.

The DFM here presented was defined on the basis of an extensive prior knowledge of the main components involved in K-ras transformed cells grown in this perturbed condition. In particular, cancer cell death processes occurring upon glucose starvation along two major pathways were considered: (i) a pathway centered on mitochondria (reactive oxygen species (ROS), Adenosine Triphosphate (ATP) depletion, calcium (Ca^{2+}) overloading) [120, 121], and (ii) an endoplasmic reticulum (ER)-stress pathway associated with reduction of *N*-glycosylation and cell attachment, and a consequent activation of the Unfolded Protein Response (UPR) leading to cell death [122, 123]. By contrast, two major mechanisms were indicated as survival routes: (i) mitochondrial activity rewiring, and (ii) autophagy. The model was validated against experimental data obtained from mouse fibroblasts transformed by oncogenic K-ras expression (NIH3T3 K-ras cells) and a human K-ras mutated breast cancer cell line (MDA-MB-231), both grown in unperturbed and different perturbed conditions (see Appendix B for additional details). Ultimately, this DFM approach aims at efficiently identifying novel therapeutic treatments that maximize apoptosis over survival in cancer cells in the above mentioned growth conditions.

8.1.1 Model definition

The model can be represented as a FN consisting of 25 nodes, corresponding to heterogeneous components—*e.g.*, proteins, small molecules and metabolites, biochemical pathways, cellular processes, output phenotypes—while edges between nodes indicate the known positive or negative regulations among these components (Figure 8.1). To describe cell death processes as a result of glucose starvation in cancer cells, the following cellular components were considered:

1. the main cellular processes and components involved in energy production, such as glucose, glycolysis, ATP, mitochondria and autophagy;

2. different mitochondrial processes and components, such as mitochondrial potential (DeltaPsi) variation, ROS generation, mitochondrial Complex I (CI) activity, and B-cell lymphoma 2 (Bcl2) expression and activity;

3. some processes and components related to the ER, such as UPR, Ca^{2+}, C/EBP-homologous protein (CHOP), and c-Jun N-terminal kinase (JNK);

4. processes and proteins involved in cellular adhesion, such as Hexosamine Biosynthesis Pathway (HBP), *N*-glycosylation, attachment, and Src;

5. proteins and processes involved in the regulation of cell death and survival mechanisms, such as Ras, Extracellular-signal-regulated kinase (ERK), Death-associated protein kinase 1 (DAPK), Bcl2, Beclin1 (BCN1), Caspase 3 (Casp3),

protein kinase A (PKA), and the phenotypes related to apoptosis, necrosis and survival of cells;

6. the protein Ras-GTP to mimic the hyperactivation of K-ras inside tumor cells displaying both the Warburg effect, and PKA as a key node involved in cancer cells survival to glucose starvation.

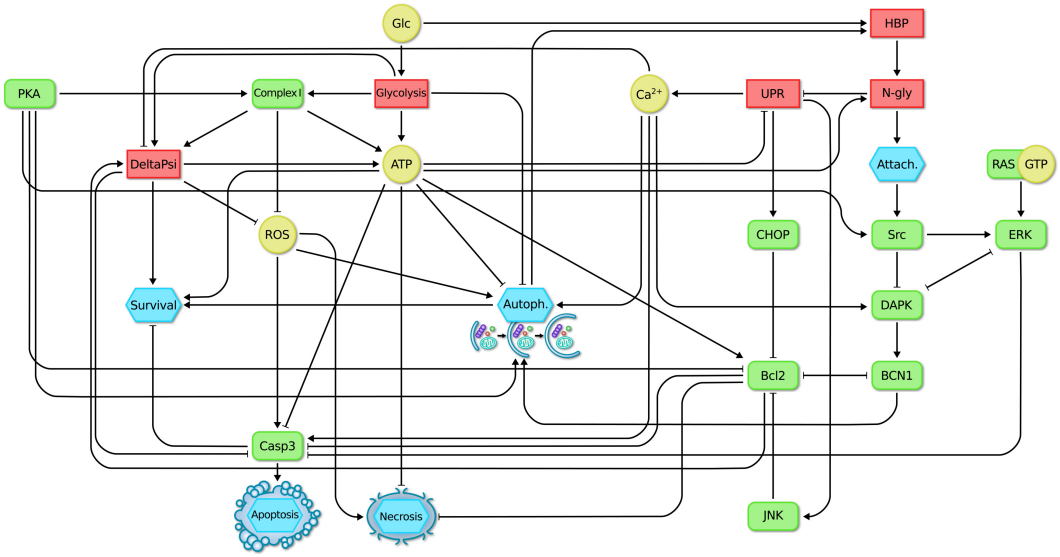


FIGURE 8.1: Interaction network of the model of cell death and survival. Yellow circles represent metabolites and ions, green rectangles represent proteins, red rectangles represent pathways or cellular processes, light blue hexagons represent the system phenotypes related to cell death. Positive and negative regulations are pictured as arrows and blunt-ended arrows, respectively. Glucose, Ras-GTP and PKA are the input variables; survival, autophagy, apoptosis and necrosis are the output variables, while the remaining are inner variables.

Since not all of these components (*e.g.*, apoptosis, survival, UPR, HBP, etc.) can be formally represented by a quantitative variable, and some interactions cannot be specified by means of kinetic reactions, fuzzy logic proves useful to handle the heterogeneity of this system and the lack of parameters by defining suitable linguistic variables representing general concepts such as concentration, activation, or presence of a component.

Literature and expert knowledge were exploited in order to formalize the interactions existing among the components considered in the model, resulting in 252 fuzzy rules. The list of the linguistic variables, together with the references employed for the definition of the model, can be found in Table 8.1. The complete list of linguistic variables and their associated linguistic terms employed to describe all the possible states of the variable (*e.g.*, High or Low concentration) are given in Table 8.2, while the set of fuzzy rules defined for each linguistic variable can be found in Appendix A. The list of output crisp values associated to each variable can be found in Table 8.3. Moreover, the complete model can be downloaded as a .fms file from `https://github.com/aresio/FUMOSO`.

TABLE 8.1: Programmed cell death model: variables, regulatory elements and literature references

Variable	**Regulatory elements**	**Reference**
Apoptosis	Caspase 3	[120, 121]
ATP	Complex I/DeltaPsi/Glycolysis	[124, 125]
Attachment	N-glycosylation	[126–128]
Autophagy	Calcium	[129–131]
	BCN1	[132]
	PKA	[112]
	ATP/Glycolysis	[131, 133]
	ROS	[134, 135]
Bcl2	CHOP/JNK	[122, 136–140]
	BCN1	[141–143]
	ATP	[144, 145]
	PKA	[146, 147]
BCN1	Bcl2	[141–143]
	DAPK	[148]
Calcium	UPR	[149]
Caspase3	ATP/Bcl2/Calcium/DeltaPsi/ROS	[120, 129, 130, 150–152]
	ERK	[153]
CHOP	UPR	[112, 122, 123, 126, 136]
Complex I	Glycolysis/PKA	[124, 125]
DAPK	ERK	[154]
	SRC	[155]
	Calcium	[156, 157]
DeltaPsi	Bcl2	[158–160]
	Complex I/Glycolysis	[120, 125, 161]
	Calcium	[150, 162]
ERK	Ras-GTP	[163]
	DAPK	[164]
	SRC	[165]
Glycolysis	Glucose	[111, 166]

TABLE 8.1 (CONTINUED): Programmed cell death model: variables, regulatory elements and literature references

Variable	Regulatory elements	Reference
HBP	Glucose	[136, 167, 168]
	Autophagy	[112, 169]
JNK	HBP	[136, 170]
Necrosis	ATP/Bcl2/ROS	[135, 171–173]
N-glycosylation	HBP	[126, 167]
PKA		[112]
Ras-GTP		[124, 125, 136, 166]
ROS	Complex I	[124, 151, 174]
	DeltaPsi	[150]
SRC	Attachment	[175–177]
	PKA	[112, 178, 179]
Survival	ATP/Autophagy/Caspase 3/DeltaPsi	[112, 133, 150, 180]
UPR	N-glycosylation	[126, 181, 182]
	ATP	[183]

TABLE 8.2: Linguistic variables, together with their linguistic terms and initial state in the programmed cell death model

Variable	**Terms**	**Initial state**
Apoptosis	Low, Medium, High	Low
ATP	Very-Low, Low, Medium, High	High
Attachment	Low, High	High
Autophagy	Low, Medium, High	Low
Bcl2	Low, Medium, High	High
BCN1	Low, Medium, High, Very High	Low
Calcium	Low, Medium, High	Low
Caspase3	Low, Medium, High	Low
CHOP	Low, Medium, High	Low
ComplexI	Less func., Medium func., More func.	Medium func.
DAPK	Low, High	Low
DeltaPsi	Low, High	High
ERK	Low, High	High
Glycolysis	Low, Medium, High	High
HBP	Low, High	High
JNK	Low, High	Low
Necrosis	Low, High	Low
N-glycosylation	Low, Medium, High	High
PKA	Low, High	Low
RAS-GTP	On, Off	On
ROS	Low, Medium, High	Medium
SRC	Low, High	High
Survival	Low, High	High
UPR	Low, Medium, High	Low

TABLE 8.3: Output crisp values of linguistic terms in the programmed cell death model

	Term	Output crisp value
ATP concentration	Very low	0.0
	Low	0.1
	Medium	0.5
	High	1.0
BCN1 concentration	Low	0.0
	Medium	0.33
	High	0.66
	Very high	1.0
Complex I activity	Less functional	0.0
	Medium functional	0.5
	More functional	1.0
Ras-GTP activation	Off	0.0
	On	1.0
State of other components	Low	0.0
	Medium*	0.5
	High	1.0

*when present

Three variables in the DFM represent the input of the system, namely, Glucose, Ras-GTP and PKA. Glucose is formally regulated by a custom input function that simulates glucose consumption. The update function $\phi_1(t)$ for the Glucose variable is defined as follows:

$$\phi_1(t) = \begin{cases} 1.0 & \text{if } t < 0.075, \\ \frac{1.0}{7\ t^{0.75}} - 0.185 & \text{if } 0.075 \leq t \leq 0.7, \\ 0.0 & \text{if } t > 0.7, \end{cases}$$

where t is the current time step in the simulation interval $[t_0, t_{max}] = [0, 1]$ (arbitrary units). Function $\phi_1(t)$ mimics the availability of glucose at the beginning of the simulation ($t < 0.075$) and its consumption by the cancer cell (when $0.075 \leq t \leq 0.7$), until its complete depletion ($t > 0.7$) from the culture medium.

The Ras-GTP variable is updated by using a constant function $\phi_2(t) = 1.0$, for all $t \in [0, 1]$, which mimics the hyperactivation of the Ras protein in cancer cells.

The PKA variable is updated by using two constant functions: $\phi_3(t) = 0.0$ and $\phi_4(t) = 1.0$, for all $t \in [0, 1]$, to mimic the normal and the hyperactivated PKA states, respectively.

8.1.2 Model validation

The DFM was experimentally validated against data obtained from cell cultures grown in different glucose availability, and in presence or absence of several protein and process-modulating molecules. Experimental data were obtained on NIH3T3 K-ras cells and a human model of breast cancer, the MDA-MB-231 cell line, carrying the oncogenic K-ras gene. The two cell lines were used in an exchangeable manner, since previously published results indicated that they have a comparable behavior as regard to metabolic and pro-survival responses, when grown in 25 mM or 5 mM glucose (high glucose, HG) or 1 mM glucose (low glucose, LG) [112, 124, 136, 184]. A detailed description of the experimental protocols employed for this work is provided in Appendix B.

The outcome of the model simulation is provided in Figure 8.2, where the dynamics of each variable are plotted against the dynamics of Glucose, the main input of the model. Altogether, these results were in strong agreement with previously published findings (see Table 8.1 for reference). Additionally, the DFM was validated against new experimental data, measured both in unperturbed and perturbed conditions.

Figure 8.3 shows the simulated dynamics of the observable output variables of the DFM, that is, survival, autophagy, apoptosis and necrosis, as well as the experimental data obtained on the MDA-MB-231 cell line, confirming the glucose-dependent model outcome in unperturbed conditions. Indeed, both the simulations and the experimental results show a decrease in cell survival (dark green line in Figure 8.3a *vs.* Figure 8.3b), associated with an increase in cell death either by apoptosis (orange line in Figure 8.3a *vs.* Casp3 activation in Figure 8.3b) or necrosis (magenta line in Figure 8.3a *vs.* High Mobility Group Box 1 (HMGB1) release in the medium in Figure 8.3b). Moreover, they show a decrease in cell adhesion (light blue line in Figure 8.3a *vs.* Figure 8.3c), associated with a significant reduction of membrane proteins *N*-glycosylation (Figure 8.3d).

Additional simulations were also performed to consider the effects of the hyperactivation of PKA. Figure 8.4 shows that the DFM correctly predicts an increase in cell survival and attachment given by the activation of PKA (Figure 8.4a, dark green and light blue lines, respectively), as validated by previous experimental results and literature data [112, 124, 185, 186]. Indeed, under glucose starvation (right side of the plot in Figure 8.4a), PKA activation—obtained by cell growth in 1 mM glucose and by a daily treatment with 10 μM forskolin (FSK), a specific activator of PKA—induces cell survival associated with increased MDA-MB-231 cell adhesion (Figure 8.4b) and increased *N*-glycosylation of membrane proteins (Figure 8.4c-d).

The DFM was also validated in perturbed conditions, namely, HBP or *N*-glycosylation inhibition by azaserine (aza) and tunicamycin (tuni), respectively, in HG availability (experimental scheme provided in Figure B.1a, Appendix B), and ATP addition in glucose starvation (experimental scheme provided in Figure B.1b, Appendix B). The simulations of HBP or *N*-glycosylation inhibition (Figure 8.5a and d)

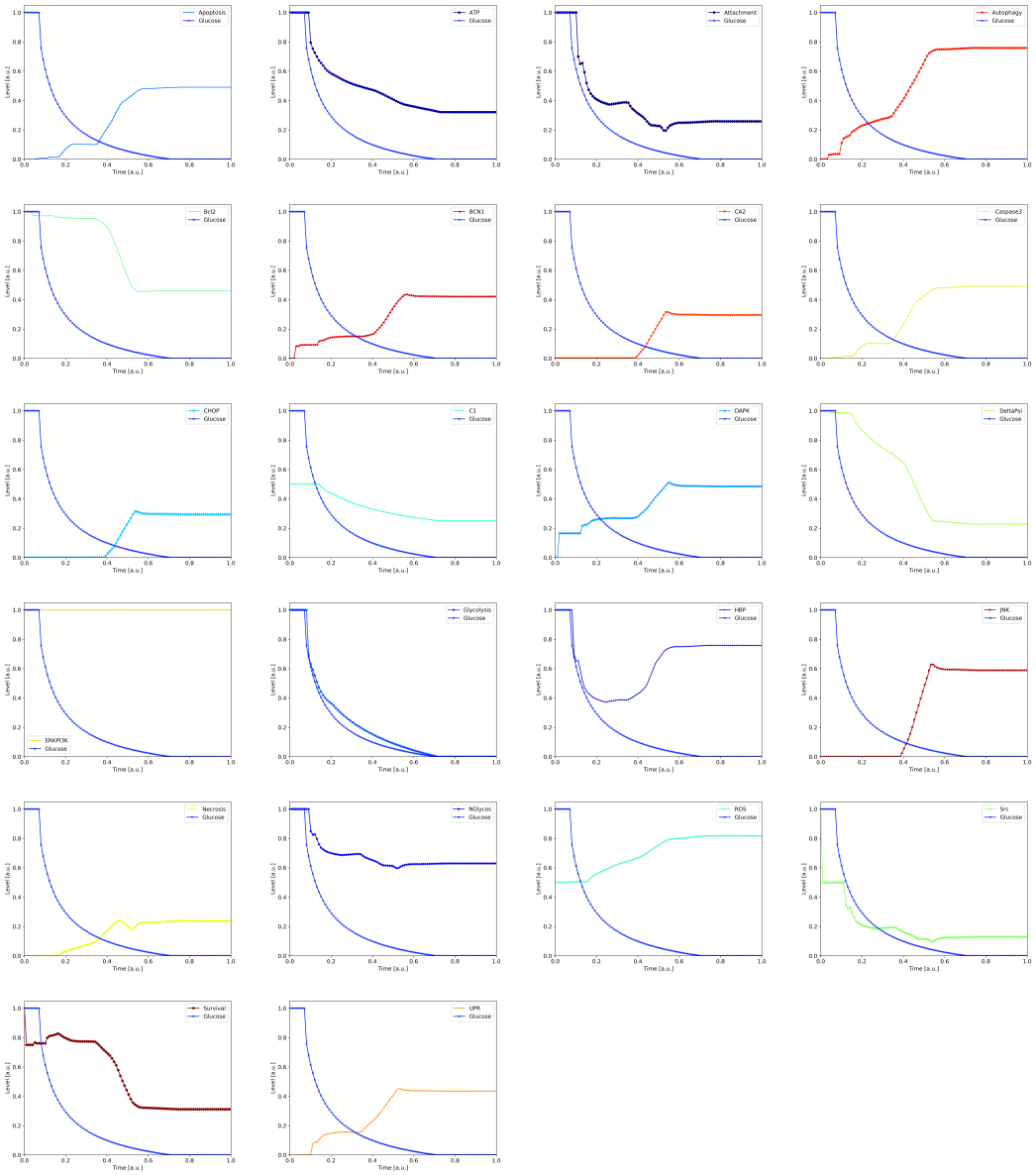


FIGURE 8.2: Simulated dynamics of the 22 regulated variables (*i.e.*, all but PKA, Ras-GTP, Glucose) of the cell death model, with respect to the function describing glucose availability.

show an increase in cell death (orange line for apoptosis and magenta line for necrosis, in Figure 8.5a, d), which is in line with the experimental data shown in Figure 8.5b, c, e and f. Conversely, 100 μM ATP added in glucose starvation recovers cell survival capabilities, similarly to 10 mM *N*-acetylglucosamine (GlcNAc), a specific HBP substrate. This was highlighted both in the DFM simulation (dark green line in Figure 8.5g), and in the experimental results, by an increase in alive cells (Figure 8.5h) and the reduction of Casp3 activation and HMGB1 release (Figure 8.5i).

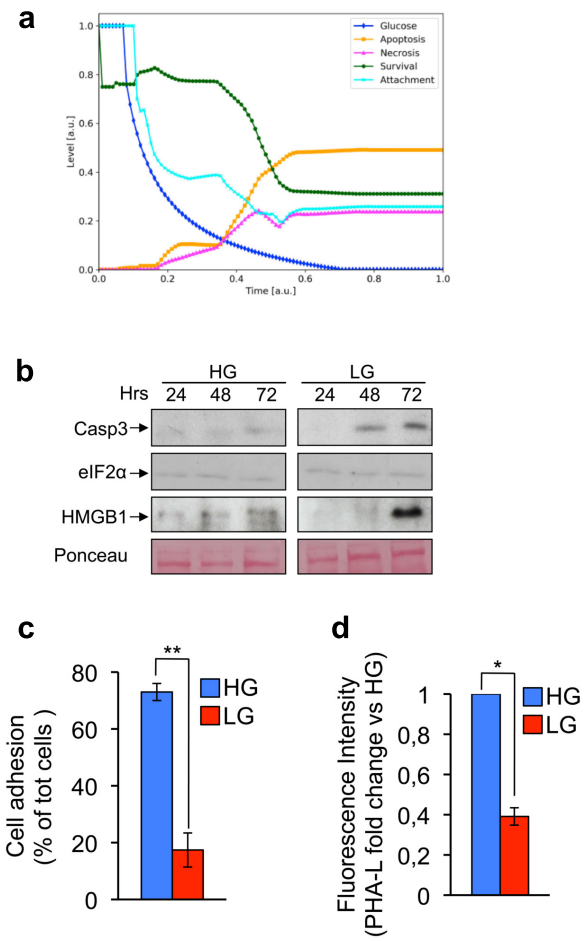


FIGURE 8.3: Validation of the model for cancer cells grown in progressive limiting amount of glucose. (a) Simulation outcome of the three main model output components (apoptosis, necrosis and survival), and of one cellular process (attachment) upon time-dependent glucose depletion. (b) Total protein extract from MDA-MB-231 cells cultured in HG (25 mM) and LG (1 mM) glucose for 72h analyzed by Western blotting at indicated time points. Apoptosis was evaluated by Casp3 cleavage, necrosis by HMGB1 release in the culture medium. Casp-3 was normalized by using eIF2α protein level, while HMGB1 secretion by a Ponceau staining of the loaded medium. (c) Cell adhesion evaluation in cells cultured for 72h in HG or LG by using Boyden chamber. (d) Fluorescence intensity quantitation of membrane *N*-linked protein levels of *Phaseolus vulgaris* (PHA-L) lectin staining cells grown as in (b). All data represent the average of at least three independent experiments ($\pm$ s.d.); *p$<$0.05, **p$<$0.01 (Student's *t*-test).

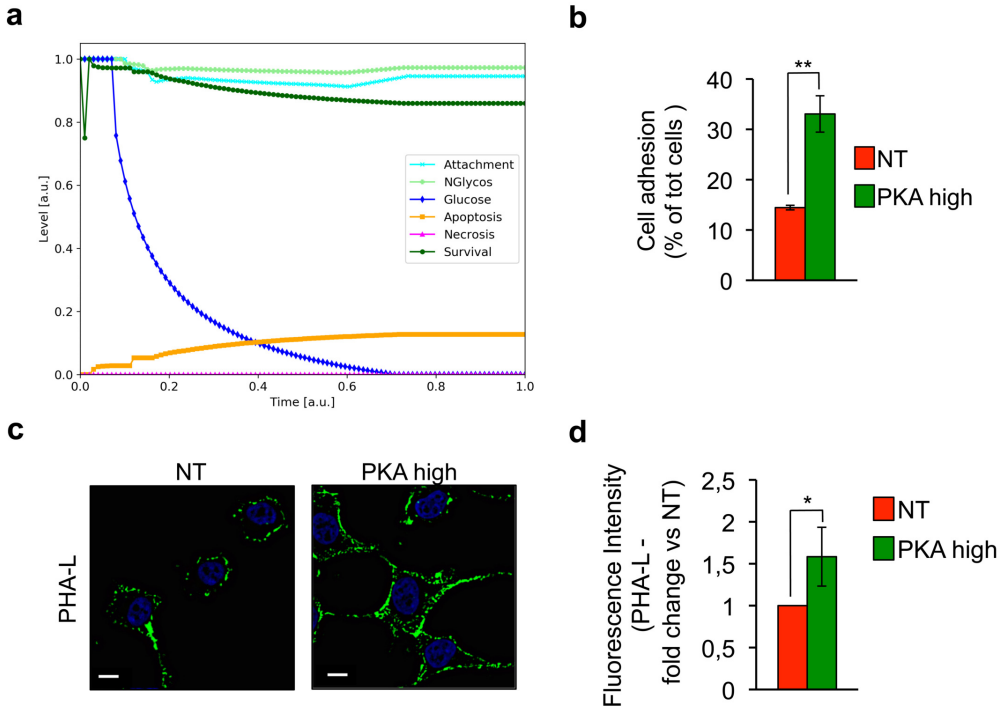


FIGURE 8.4: Validation of the model for cancer cells grown in progressive limiting amount of glucose upon PKA hyperactivation (PKA in High state). (a) Simulation outcome of the three main model output (apoptosis, necrosis and survival), and of one cellular process (attachment), upon time-dependent glucose depletion and PKA hyperactivation. (b) MDA-MB-231 cells were cultured for 72h in LG and treated or not with FSK (daily treatment with 10 μm) to mimic *in vivo* the PKA hyperactivation, and then analyzed for cell adhesion. (c) Membrane *N*-linked protein levels of cells grown as in (a) by confocal microscopy of PHA-L lectin staining (40$\times$ magnification, 10 μm scale), and (d) relative fluorescence intensity quantitation. All data represent the average of at least three independent experiments ($\pm$ s.d.); *p$<$0.05, **p$<$0.01 (Student's *t*-test).

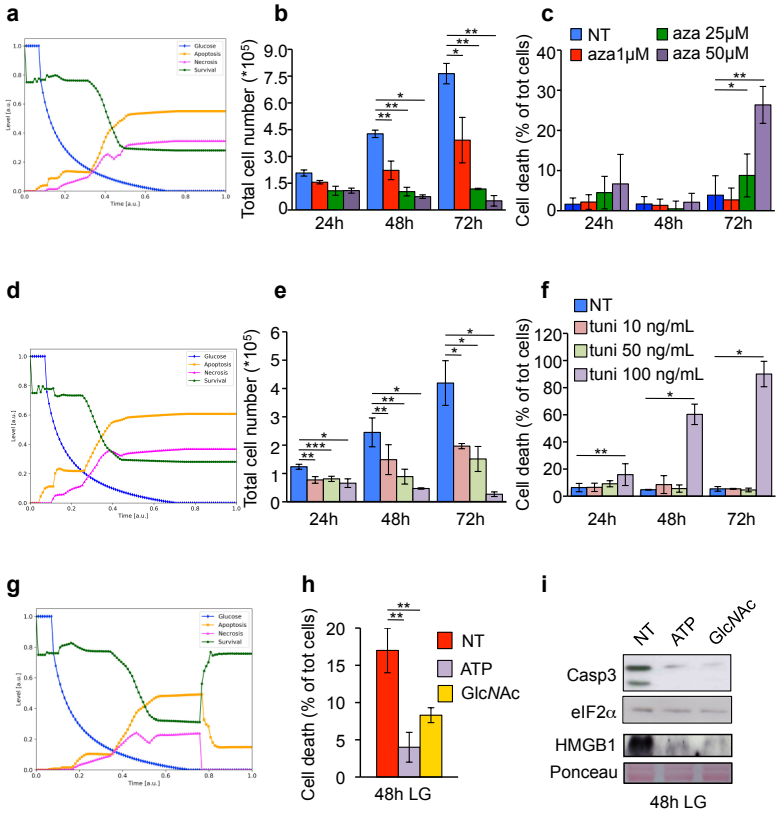


FIGURE 8.5: Validation of the model for cells that underwent HBP or *N*-glycosylation inhibition, and ATP addition. (a) Simulation outcome of the three model outputs (apoptosis, necrosis and survival) upon HBP inhibition at time $t = 0.0$. (b)-(c) MDA-MB-231 cells were grown in HG and in presence of increasing doses of the HBP inhibitor aza (single treatment at time 0h), in order to evaluate either (b) cell proliferation or (c) cell death. The analysis was performed by Trypan Blue Exclusion Method; the experimental scheme is shown in Figure B.1a. (d) Simulation outcome of the three model outputs (apoptosis, necrosis and survival) upon *N*-glycosylation inhibition at time $t = 0.0$. (e)-(f) MDA-MB-231 cells were grown in HG and in presence of increasing doses of the *N*-glycosylation inhibitor tuni (single treatment at time 0h), in order to evaluate either (e) cell proliferation or (f) cell death. The experimental scheme is shown in Figure B.1a. The analysis was performed as in (b) and (c). (g) Simulation outcome of the three model outputs (apoptosis, necrosis and survival) upon ATP node increase at time $t = 0.75$, when the glucose in the medium is 0 mM, as previously experimentally evaluated [124]. (h) MDA-MB-231 cells were grown in LG for 48h and then treated for 24h with 100 μM ATP (single treatment at time 48h) or 10 mM Glc*N*ac, in order to evaluate cell death. The experimental scheme is shown in Figure B.1b. The analysis was performed as in (b) and (c). (i) Total protein extracts from cells grown as in (h) were analyzed by Western blotting to evaluate induction of apoptosis by Casp3 cleavage, and necrosis by HMGB1 release in the culture medium. Representative images of at least three independent experiments are shown. All data represent the average of at least three independent experiments ($\pm$ s.d.); *p<0.05, **p<0.01, ***p<0.001 (Student's *t*-test).

8.1.3 Perturbation analysis

The coupling of the DFM simulation by means of FUMOSO and a global optimization algorithm, namely SA, allowed to automatically search for and detect the combinations of perturbations that maximize pro-apoptotic processes in cancer cells, to the purpose of guiding the development of novel therapeutic treatments. Indeed, apoptosis, unlike necrosis, does not induce inflammation in the surrounding tissues, thus it is a preferred outcome during the treatment of cancer cells.

Accordingly, the objective function used in this work aims at maximizing the level of Apoptosis while minimizing the level of Survival, by perturbing the minimal number of model components. Given a perturbation π generated by SA, the objective function is defined as:

$$F(\pi) = \frac{[s^{\pi}_{Apoptosis}(t_b + \delta) - s^{\pi}_{Apoptosis}(t_b)] - [s^{\pi}_{Survival}(t_b + \delta) - s^{\pi}_{Survival}(t_b)]}{\delta \cdot |\pi|},$$

where $s^{\pi}_{Apoptosis}(\cdot)$ and $s^{\pi}_{Survival}(\cdot)$ are the numerical values of the Apoptosis and Survival linguistic variables, respectively, at the indicated time steps, obtained in a simulation where π is applied, t_b is the time step at which the perturbation π is applied, $\delta > 0$ is a sampling time instant, that is, the time delay after which the effect of the perturbation is evaluated (with $t_b + \delta \leq t_e$, being $t_e \leq t_{max}$ the end of the perturbation π), and $|\pi|$ is the number of linguistic variables that are perturbed in π. A graphical representation of the evaluation of the objective function is given in Figure 8.6.

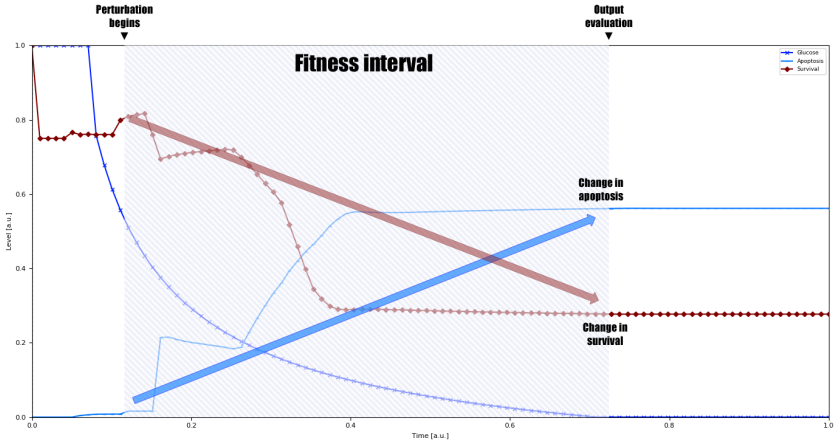


FIGURE 8.6: Evaluation of the objective function used by Simulated Annealing. The evaluation interval (*i.e.*, $t_b + \delta$) is highlighted in gray.

The results showed in this work were obtained by setting $t_b = 0$, $t_e = t_{max}$, and $\delta = 0.13$ in all tests. These values were chosen in order to match the experimental protocols, that is, the perturbation is carried out by some modulating molecules that persist in the culture medium until glucose depletion, while cell death is measured when the amount of glucose is approximately halved. The starting temperature of SA was set to $T_0 = 0.1$, linearly decreasing to 0 during the iterations, and the number

of perturbed variables per iteration was set to $e = 2$. This value of e ensured the generation of combination of perturbations, while at the same time minimizing the number of perturbed variables. Lastly, since some perturbations are known to be unfeasible due to technical or biological reasons, a subset of all the possible output crisp values of the system was selected to perform the optimization; this subset is listed in Table 8.4.

TABLE 8.4: Model components and corresponding states that were perturbed during SA optimization

Model component	**States**
Attachment	Unperturbed
	Low
	High
Autophagy	Unperturbed
	Low
	Medium
	High
Bcl2	Unperturbed
	Low
	Medium
	High
Calcium	Unperturbed
	Low
	Medium
	High
CHOP	Unperturbed
	Low
	Medium
	High
Complex I	Unperturbed
	Less functional
	Medium functional
	More functional
DAPK	Unperturbed
	Low
	High
DeltaPsi	Unperturbed
	Low
	High
ERK	Unperturbed
	Low
	High

TABLE 8.4 (CONTINUED): Model components and corresponding states that were perturbed during SA optimization

Model component	States
HBP	Unperturbed
	Low
	Medium
	High
JNK	Unperturbed
	Low
	High
N-glycosylation	Unperturbed
	Low
	Medium
	High
ROS	Unperturbed
	Low
	Medium
	High
SRC	Unperturbed
	Low
	High
UPR	Unperturbed
	Low
	Medium
	High

Since SA is a stochastic algorithm, different runs generally yield different solutions, still commonly converging to the optimal perturbations that represent the attractors of the global optimization. The optimization analysis was carried out both in the case of low and hyperactivated states of PKA, for a total of 30 runs each, in order to collect a set of different perturbations. The resulting list of perturbations found by SA—validated either in this work or confirmed by previous experimental evidences—is given in Table 8.5 and Table 8.6 for the PKA Low and High conditions, respectively, ranked by their objective function value in descending order. From these, a subset of promising single/double perturbations was selected and tested in laboratory to assess their effectiveness in inducing death by apoptosis in MDA-MB-231 cells.

TABLE 8.5: Perturbations found by SA with PKA set to the Low state

Rank	Variable	State	Reference
1	DeltaPsi	Low	[187–189]
2	UPR	High	This work
3	DeltaPsi UPR	Low High	[190, 191]
4	Complex I	Less functional	This work
5	Bcl2	Low	[192]
6	UPR Autophagy	High Low	This work
7	Complex I Autophagy	Less functional Low	This work
8	N-glycosylation HBP	Low Low	This work
9	N-glycosylation Autophagy	Low Low	This work

TABLE 8.6: Perturbations found by SA with PKA set to the High state

Rank	Variable	State	Reference
1	UPR	High	This work
2	DeltaPsi	Low	[187–189]
3	DeltaPsi UPR	Low High	[190, 191]
4	ROS UPR	High High	[190, 191]
5	N-glycosylation Autophagy	Low Low	This work
6	N-glycosylation HBP	Low Low	This work

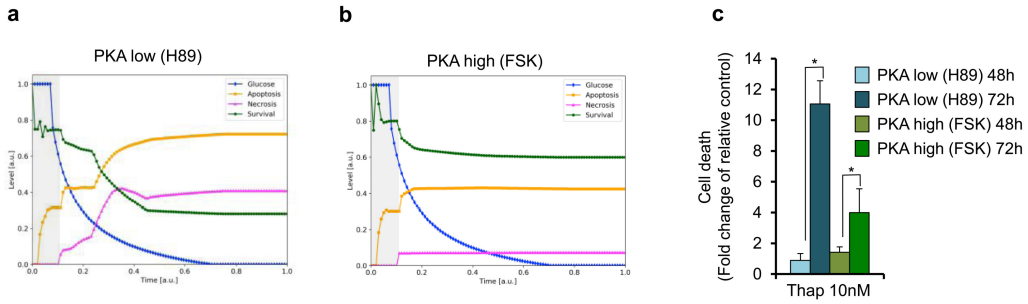


FIGURE 8.7: Assessment of the effects of perturbations (UPR activation) predicted by the global optimization algorithm. (a)-(b) Simulation outcome of the three model outputs (apoptosis, necrosis and survival) upon UPR activation, either in (a) PKA Low state or (b) PKA High state. The perturbation was applied from time $t_b = 0$ to the end of the simulation, and evaluated after $\delta = 0.13$ a.u. (shaded area). (c) MDA-MB-231 cells, grown in HG, were daily treated with 10 μM FSK mimicking the PKA High state, or 5 μM H89 mimicking the PKA Low state and, upon 24h, also with 10 nM thap (single treatment). Samples were evaluated for cell death at 48h and 72h post-treatment by using Trypan Blue Exclusion Method. The experimental scheme is shown in Figure B.1c. All data represent the average of at least three independent experiments ($\pm$ s.d.); *p$<$0.05 (Student's t-test).

Figure 8.7 and Figure 8.8 show the comparison between the simulated dynamics and the experimental data for *single* perturbations predicted by SA. In particular, Figure 8.7 shows the effects of UPR activation achieved by using 10 nM thapsigargin (thap), while Figure 8.8 shows the effects of CI inhibition achieved by using 10 nM rotenone (rot) or 20 nM piericidin (pier) in high and low glucose availability, respectively (the experimental scheme for both perturbations is shown in Figure B.1c, Appendix B). Both perturbations were chosen taking into account their rank and the existence of previous data indicating that the pharmacological over-activation of the UPR pathway, as well as CI inhibition, can lead to cancer cell death [193, 194]. For both perturbations, high or low activation of PKA was also experimentally analyzed. In particular, to avoid the endogenous activation of PKA under the experimental conditions, low PKA was achieved by cell treatment with a known PKA inhibitor, H89 [195] (experimental details are available in Figure B.1, Appendix B).

Notably, the model properly predicted the experimental data. Indeed, an enhanced cell death was observed in the simulations (orange line for apoptosis and magenta line for necrosis in Figure 8.7a and b) as well as in the experimental data upon both treatments (Figure 8.7c). This increase is evident when moving from a high availability of glucose (left side of the simulation plots) to a situation of glucose starvation (right side of the simulation plots), and especially in the PKA Low state with respect to the PKA High state, suggesting an important role of PKA in cancer cell survival in acute UPR activation or in glucose starvation. These computational results are consistent with the experimental measurements, as represented by the graph bars in Figure 8.7c, which show a higher level of cell death in glucose starvation (72h), with the fold change being higher when PKA is not activated. Similar considerations can be done for chronic CI inhibition, as shown in Figure 8.8.

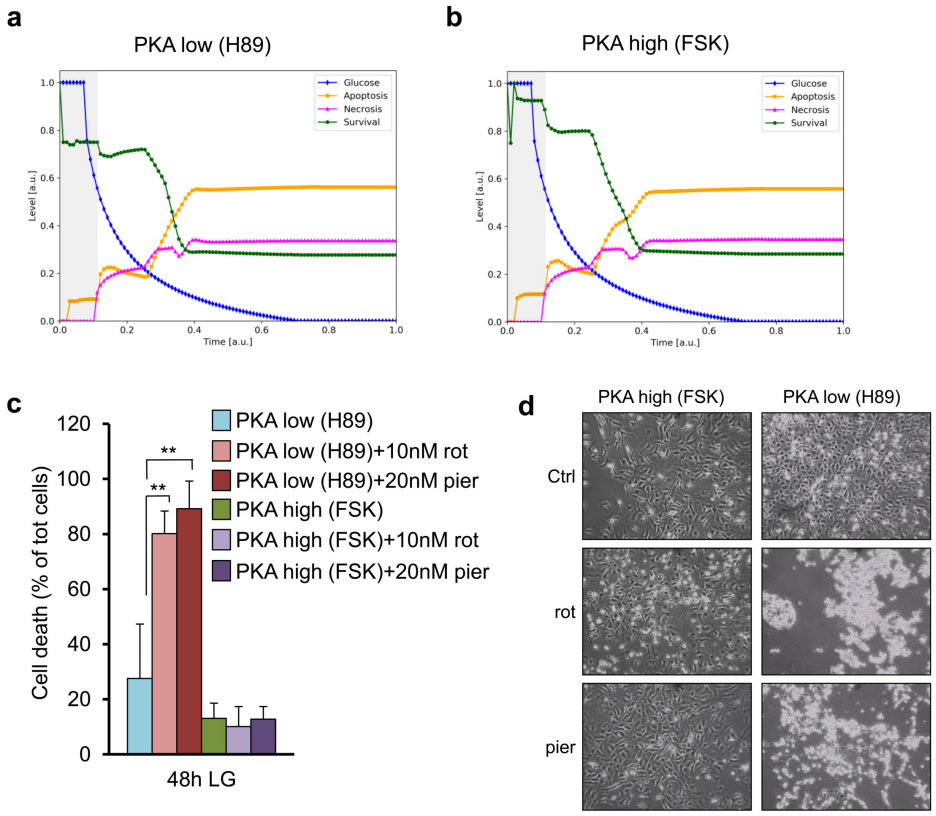


FIGURE 8.8: Assessment of the effects of perturbations (CI inhibition) predicted by the global optimization algorithm. (a)-(b) Simulation outcome of the three model outputs (apoptosis, necrosis and survival) upon Complex I inhibition, either in (d) PKA Low state or (e) PKA High state. The perturbation was applied from time $t_b = 0$ to the end of the simulation, and evaluated after $\delta = 0.13$ a.u. (shaded area). (c) MDA-MB-231 cells were grown in LG for 48h and then treated for 24h with 10 nM rotenone or 20 nM piericidin (single treatment at time 48h), in order to evaluate cell death by using Trypan Blue Exclusion Method. The experimental scheme is shown in Figure B.1c. (d) Representative morphological images of MDA-MB-513 231 cells treated as in (c). The experimental scheme is shown in Figure B.1c. All data represent the average of at least three independent experiments ($\pm$ s.d.); *p$<$0.05, **p$<$0.01 (Student's *t*-test).

Analogously, Figure 8.9 shows the comparison between the simulated dynamics and the experimental data for the *double* perturbations predicted by SA. In particular, it shows the effects of UPR activation coupled with autophagy inhibition (Figure 8.9a-c), *N*-glycosylation and HBP inhibition (Figure 8.9d-f), *N*-glycosylation and autophagy inhibition (Figure 8.9g-i). Both the simulation outcomes and the experimental data show an increase in cell death, especially when moving towards a state of glucose starvation, and the protective role of PKA. This result is evident by comparing, for example, the final states reached in each condition by apoptosis (orange line) and necrosis (magenta line) in Figures 8.9a, d and g, with the states reached by the same variables in Figure 8.9b, e and h. These computational results are consistent with the experimental measurements shown by the graph bars in Figures 8.9c, f, and i.

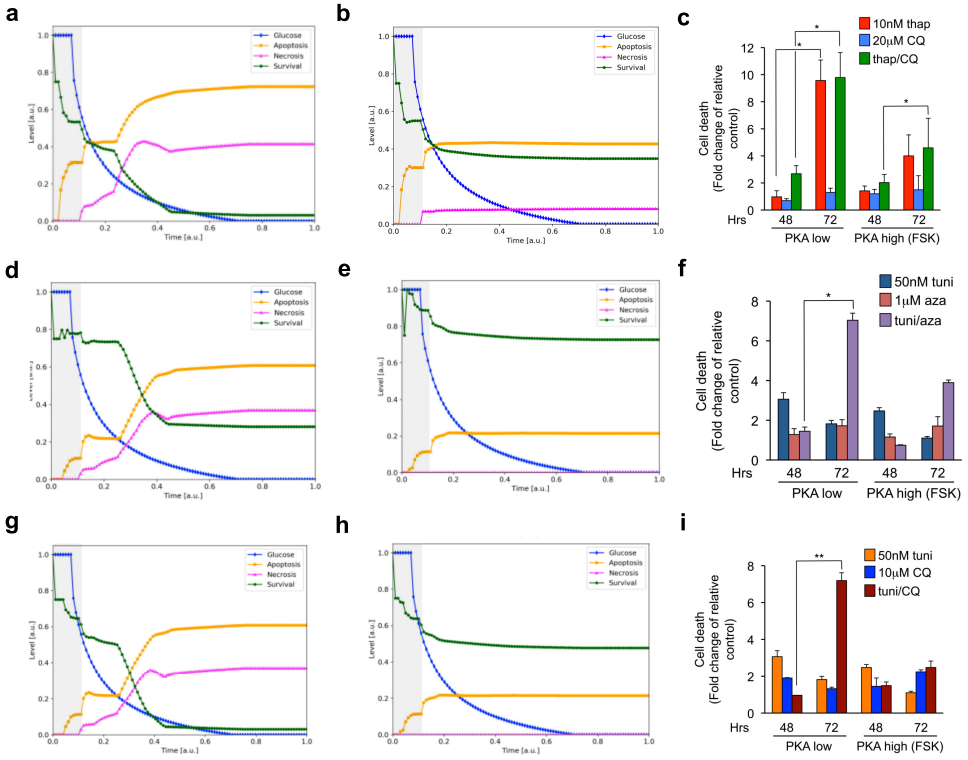


FIGURE 8.9: Assessment of the effects of double perturbations predicted by global optimization. (a)-(b) Simulation outcome of the three model outputs (apoptosis, necrosis and survival) upon UPR activation and autophagy inhibition, either in (a) PKA Low state or (b) PKA High state. The perturbation was applied from time $t_b = 0$ to the end of the simulation, and evaluated after $\delta = 0.13$ a.u. (shaded area). (c) MDA-MB-231 cells, grown in HG, were daily treated with 10 μM FSK mimicking the PKA High state and, upon 24h, also with 10 nM thap and 20 μM chloroquine (CQ) (single treatment of both). Samples were evaluated for cell death at 48h and 72h post-treatment by using Trypan Blue Exclusion Method. The experimental scheme is shown in Figure B.1d. (d)-(e) Simulation outcome of the three main model output (apoptosis, necrosis and survival) upon HBP and *N*-glycosylation inhibition, either in (d) PKA Low state or (e) PKA High state. (f) MDA-MB-231 cells, grown in HG, were daily treated with 10 μM FSK mimicking the PKA High state and, upon 24h, also with 1 μM aza and 50 ng/mL tuni (single treatment of both). Samples were evaluated for cell death at 48h and 72h post-treatment by using Trypan Blue Exclusion Method. The experimental scheme is shown in Figure B.1d. (g)-(h) Simulation outcome of the three model outputs (apoptosis, necrosis and survival) upon *N*-glycosylation and autophagy inhibition, either in (g) PKA Low state or (h) PKA High state. (i) MDA-MB-231 cells, grown in HG, were daily treated with 10 μM FSK mimicking the PKA High state and, upon 24h, also with 50 nM tuni and 10 μM CQ (single treatment of both). Samples were evaluated for cell death at 48h and 72h post-treatment by using Trypan Blue Exclusion Method. The experimental scheme is shown in Figure B.1d. All data represent the average of at least three independent experiments ($\pm$ s.d.); *p<0.05, **p<0.01 (Student's *t*-test).

To analyze the combinatory effect on cell survival, when possible, sub-toxic concentrations of the compounds were employed, previously determined experimentally on this cell model (see, *e.g.*, tuni and aza). In addition, in these set of validation

experiments, it was also evaluated whether the model could predict the endogenous PKA behavior, since no H89 inhibitor was used (experimental scheme in Figure B.1d, Appendix B). Combination effects of selected compound pairs exceeded the effects of single compounds for the majority of combinations, and well fitted with the model outcome also in absence of exogenous PKA inhibition. Thus, these results corroborated the capability of the DFM, coupled with the optimization algorithm, in predicting the system's response in perturbed conditions.

It is worthy of note that the validity of this modeling approach is supported by its performance in predicting not only recognized cell death-inducing stimuli, but also unrecognized stimuli equally leading to cell death. For instance, a sub-toxic amount of tuni, coupled with both a sub-toxic amount of the HBP inhibitor aza or the autophagic inhibitor chloroquine, enhances cancer cell death. Remarkably, both treatments are strongly attenuated by exogenous PKA stimulation, implying the involvement of this pathway in ER stress response, at least in these experimental conditions. Confirming results were shown also about the protective role of PKA upon ER stress induction by thapsigargin that, in combination with PKA inhibition, induces cancer cell death, an effect that is strongly impaired by exogenous activation of PKA. Previous data indicated that PKA activation protects cancer cells from death induced by glucose starvation [112, 124]. Here, it is shown that PKA has a protective role in cancer cells under acute ER stress. While this protective mechanism has been shown to be active in mouse embryonic fibroblasts [196] and hepatocytes [197], it has never been described in cancer cells, a fact that further supports the predictive value of DFMs coupled with optmization algorithms. Interestingly, also simultaneous CI and PKA inhibition induces increased cancer cell death, an effect that is prevented by PKA activation. Therefore, the model also predicted that PKA activation is involved in mitochondrial CI function and, more in general, in oxidative phosphorylation activity, corroborating previous results [198–200]. Altogether, these results provide the first evidence of a protective role for PKA against several treatments mimicking cellular stress conditions, such as ER stress, *N*-glycosylation inhibition, mitochondrial CI inhibition, and glucose starvation in cancer cells. These findings will potentially pave the way for the use of new PKA inhibitors in cancer therapy.

8.2 A hybrid model of biochemical signaling

To prove the feasibility and the advantages of FuzzX, this section presents the definition and analysis of a hybrid model of a biochemical signaling pathway, the Ras/cAMP/PKA pathway in the yeast *S. cerevisiae*, that is characterized by complex nonlinear behaviors arising from several feedback regulations among the system components. In particular, FuzzX is here applied to redefine, in terms of a hybrid model, a stochastic mechanistic model of this pathway first proposed in [108]. The results obtained on the hybrid model are compared to the ones obtained in the original, fully-mechanistic version of the model. The results show that, albeit mechanistic interactions are substituted by a set of expert-defined fuzzy rules, the hybrid model is able to reproduce the emergent behaviors of the system, such as the presence of a transient phase and the establishment of stable or damped oscillations.

A schematic representation of the main components of the hybrid model is depicted in Figure 8.10. The Ras/cAMP/PKA pathway consists in an intracellular cascade of biochemical reactions that are involved in the regulation of metabolism and cell cycle, in response to extracellular nutrients and stress conditions [201–203]. In particular, this pathway affects the activity of more than 90% of all yeast genes that are regulated by glucose. This control is carried out through the activation of a key protein, called PKA, which is able to regulate several downstream target proteins. In what follows, C denotes the catalytic subunit (*i.e.*, the active form) of PKA. The activation of PKA is mediated by the chemical bond with cAMP, a so called "second messenger" molecule acting as a glucose-signal transducer, which is synthesized by protein Cyr1 and degraded by protein Pde1. Cyr1 activity is controlled, in turn, by protein Ras2 that can exist in the inactive state, when it is bound to molecule GDP (*i.e.*, Ras2-GDP), or in the active state, when it is bound to molecule GTP (*i.e.*, Ras2-GTP). The switching between the inactive and active states is regulated by two proteins, called Cdc25 and Ira2.

Proteins Cdc25, Ira2 and Pde1 are regulated by PKA by means of positive and negative feedback mechanisms. These feedback controls ensure, under certain conditions, the establishment of a stable oscillatory regime in the dynamics of some pivotal molecular species in this pathway, that is, Ras2-GTP, cAMP and C. Indirect evidence of the oscillatory regimes of these species were observed *in vivo* by considering downstream proteins [204, 205], while the role of feedback regulations was thoroughly analyzed in different conditions by means of mechanistic models [108, 206, 207]. A complete definition of the original model presented in [108] and employed in the comparisons shown in the following sections, can be found in Appendix C.

8.2.1 The mechanistic module

The mechanistic module of the hybrid model is formalized as a RBM (see Section 4.1) whose main components are highlighted in blue in Figure 8.10. In particular, the

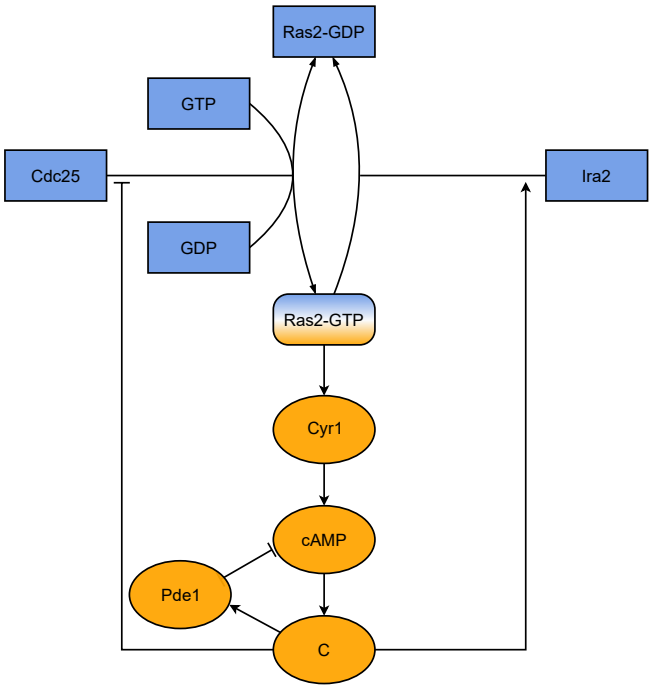


FIGURE 8.10: Diagram of the interactions among the main components of the hybrid model of the Ras/cAMP/PKA pathway. Variables belonging to the mechanistic module are represented by blue squares, variables belonging to the fuzzy module are represented by orange ellipses, while the interface species (Ras2-GTP) is represented as blue/orange rounded rectangle. Positive and negative regulations are pictured as arrows and blunt-ended arrows, respectively.

RBM considered in this work consists in the first 10 reactions of the original mechanistic model presented in [108, 206, 207] (see Table C.2, Appendix C). These reactions describe the switch cycle of the Ras2 protein between its inactive (Ras2-GDP) and active (Ras2-GTP) state, and its regulation by the proteins Cdc25 and Ira2. The set $\mathcal{P}$ of reactions of the mechanistic module, together with their kinetic parameters, is given in Table 8.7. The initial amounts of the chemical species of the mechanistic module are listed in Table 8.8 (expressed as number of molecules per cell, hence $\mathcal{X}_M = \mathbb{N}$); Table 8.8 reports also the initial values of the other variables appearing in the hybrid model, as described in what follows.

In this application of FuzzX, the dynamics of the mechanistic module is obtained by means of the "Direct Method" of SSA [67] (see Section 4.5). Since FuzzX proceeds by steps of length Δ (see Figure 7.2), during each execution of the function $U^{\mathtt{M}}$ the maximum simulation time of SSA was set to Δ.

In the hybrid version of the Ras/cAMP/PKA model, the variable corresponding to the Ras2-GTP chemical species belongs to the set $\mathcal{V}^{\mathtt{M}} \cap \mathcal{V}^{\mathtt{F}}$, and the kinetic parameters c_1, c_8 and c_{10} belong to the set $\theta^{\mathtt{M}} \cap \mathcal{V}^{\mathtt{F}}$. Together, these four elements constitute the interface $\mathcal{I}$ of the hybrid model. It should be noted that the values of the stochastic constants that do not belong to the interface never change during the simulation of the hybrid model, that is, they remain fixed to the values given in Table 8.7. On the contrary, the values of the parameters c_1, c_8 and c_{10} can be modified by the fuzzy

TABLE 8.7: Mechanistic module of the hybrid model

	Reaction	Stochastic constant
P_1:	Ras2-GDP + Cdc25 → Ras2-GDP-Cdc25	$c_1 = 1.0$*
P_2:	Ras2-GDP-Cdc25 → Ras2-GDP + Cdc25	$c_2 = 1.0$
P_3:	Ras2-GDP-Cdc25 → Ras2-Cdc25 + GDP	$c_3 = 1.5$
P_4:	Ras2-Cdc25 + GDP → Ras2-GDP-Cdc25	$c_4 = 1.0$
P_5:	Ras2-Cdc25 + GTP → Ras2-GTP-Cdc25	$c_5 = 1.0$
P_6:	Ras2-GTP-Cdc25 → Ras2-Cdc25 + GTP	$c_6 = 1.0$
P_7:	Ras2-GTP-Cdc25 → Cdc25 + Ras2-GTP	$c_7 = 1.0$
P_8:	Ras2-GTP + Cdc25 → Ras2-GTP-Cdc25	$c_8 = 1.0$*
P_9:	Ras2-GTP + Ira2 → Ras2-GTP-Ira2	$c_9 = 0.01$
P_{10}:	Ras2-GTP-Ira2 → Ras2-GDP + Ira2	$c_{10} = 0.25$*

*Kinetic parameters belonging to the interface of the hybrid model

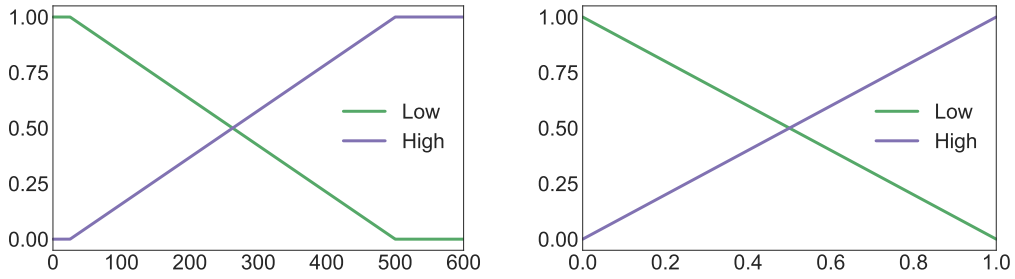
TABLE 8.8: Initial amounts of the hybrid model

	Variable	Initial value
Mechanistic module	Ras2-GDP	20,000
	Cdc25	300
	GDP	1.5×10^6
	GTP	5.0×10^6
	Ira2	200
	Ras2-GDP-Cdc25	0
	Ras2-GTP-Cdc25	0
	Ras2-Cdc25	0
	Ras2-GTP-Ira2	0
Interface	Ras2-GTP	0
Fuzzy module	Cyr1	0.0
	cAMP	0.0
	C	0.0
	Pde1	0.0

inference operated by U^{F}, according to the fuzzy rules defined in the fuzzy module, as specified in the next section.

8.2.2 The fuzzy module

The fuzzy module of the hybrid model consists in a set $\mathcal{V}^{\mathrm{F}}$ of 8 linguistic variables and a set $\mathcal{R}$ of 16 fuzzy rules; its main components are highlighted in orange in Figure 8.10.



(a) (b)

FIGURE 8.11: Membership functions of the linguistic variables: (a) Ras2-GTP; (b) Cyr1, cAMP, C, and Pde1.

The fuzzy rule base adopted in this model is a 0-order Sugeno inference system [45]. Since three linguistic variables of $\mathcal{V}^{\mathrm{F}}$—namely, c_1, c_8 and c_{10}—appear only in the consequent of the fuzzy rules as singletons, no fuzzy sets are specified for them. The fuzzy sets of all other linguistic variables, along with their associated linguistic terms, are shown in Figure 8.11a (for the linguistic variable Ras2-GTP) and Figure 8.11b (for the linguistic variables Cyr1, cAMP, C, and Pde1). Specifically, according to domain knowledge, it was assumed that the universe of discourse of Ras2-GTP is equal to $[0, +\infty)$, and its amount is Low when it is equal to or smaller than 25 molecules, while it is High when it is equal to or larger than 500 molecules. For the sake of providing a simple and cost-effective model with as few parameters as possible, it is assumed that the universe of discourse of all other variables is equal to $[0, 1]$, partitioned into two triangular fuzzy sets (Low and High).

At the end of each Δ-long simulation of the mechanistic module $\mathcal{M}$, one fuzzy inference step is performed to update the values of the variables in $\mathcal{V}^{\mathrm{F}}$. The set $\mathcal{R}$ of fuzzy rules and their output crisp values are given in Tables 8.9 and 8.10, respectively.

The fuzzy rules describe the following processes:

- the activation of protein Cyr1 (rules 1 and 2) according to the amount of Ras2-GTP;
- the production of cAMP (rules 3 and 4) according to the regulation of Cyr1;
- the activation of the catalytic subunit C of PKA (rules 5 and 6);
- the activation of Pde1 and its feedback control on cAMP (rules 7 to 10);
- the regulation that C exerts on the kinetic parameters of the mechanistic module (rules 11 to 16).

The amount of the interface variable Ras2-GTP is updated by the mechanistic module, and gets fuzzified in the fuzzy module according to its membership functions. This, in turn, determines the activation of Cyr1 that regulates all the other variables in the fuzzy module. Thus, the dynamic of the fuzzy module is driven

TABLE 8.9: Rules of the fuzzy module

No.	Rule
1	IF Ras2-GTP is Low THEN Cyr1 is Low
2	IF Ras2-GTP is High THEN Cyr1 is High
3	IF Cyr1 is Low THEN cAMP is Low
4	IF Cyr1 is High THEN cAMP is High
5	IF cAMP is Low THEN C is Low
6	IF cAMP is High THEN C is High
7	IF C is Low THEN Pde1 is Low
8	IF C is High THEN Pde1 is High
9	IF Pde1 is Low THEN cAMP is High
10	IF Pde1 is High THEN cAMP is Low
11	IF C is Low THEN c_1 is High-c_1
12	IF C is High THEN c_1 is Low-c_1
13	IF C is Low THEN c_8 is High-c_8
14	IF C is High THEN c_8 is Low-c_8
15	IF C is Low THEN c_{10} is Low-c_{10}
16	IF C is High THEN c_{10} is High-c_{10}

TABLE 8.10: Output crisp values

Term	Crisp value
Low	0.0
High	1.0
Low-c_1	0.1
High-c_1	1.0
Low-c_8	0.1
High-c_8	1.0
Low-c_{10}	0.25
High-c_{10}	2.5

by the upstream contribution of the mechanistic module. The last six fuzzy rules determine the values of the stochastic constants c_1, c_8, and c_{10}. Since the stochastic constants are used to calculate the propensity functions, which ultimately determine the probabilities of the reactions to occur, the regulation of the fuzzy module has a relevant impact onto the mechanistic module, closing the control loop between the quantitative and the qualitative regimes.

8.2.3 Analysis of the hybrid model

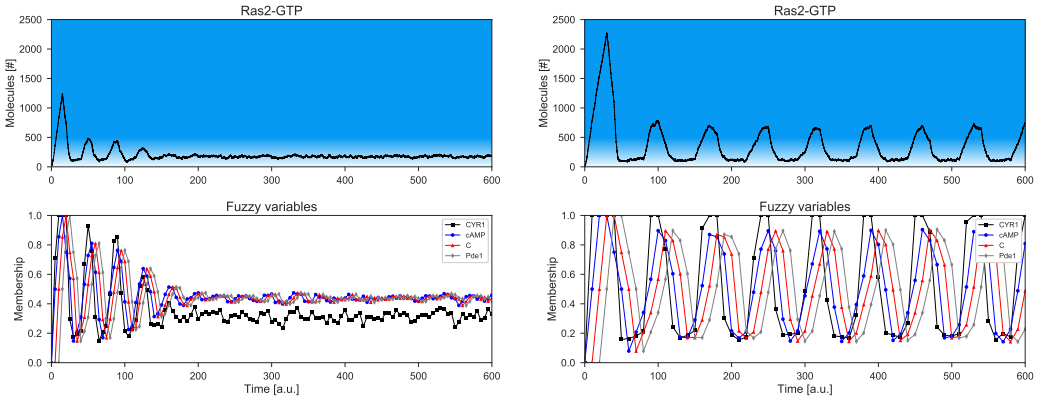
The hybrid model of the Ras/cAMP/PKA pathway has been simulated, both in unperturbed and perturbed conditions, in order to verify if the newly defined hybrid model is capable of accurately describing the same emergent behaviors observed in the original model [108]. In particular, as a first step, the effects of the choice of Δ over the dynamics were assessed. Then, the model was simulated and analyzed in normal and perturbed conditions. Deterministic simulations were also performed to show that, in FuzzX, different formalisms can be employed to define the mechanistic module.

All tests were executed on a workstation equipped with an Intel Core 7700HQ CPU with 2.80 GHz clock frequency, 16GB of RAM, and running the Operating System Microsoft Windows 10. For the analysis of the Ras/cAMP/PKA hybrid model, the mechanistic and fuzzy modules were also implemented with Python 2.7.16 and numpy, with the addition of Simpful (Chapter 6) to support the definition of fuzzy sets and fuzzy rules.

The choice of Δ and stochastic simulations

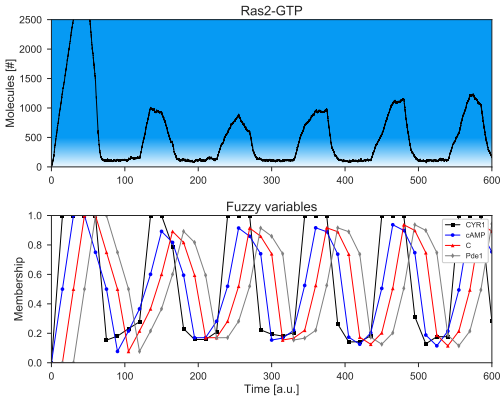
The impact of the value of the time step Δ on the overall dynamics of the Ras/cAMP/PKA hybrid model was investigated by simulating the model with values of Δ equal to 5, 10, 15, 30. These results are reported in Figure 8.12, which shows a comparison of the simulated dynamics of the interface species Ras2-GTP, and of the linguistic variables Cyr1, cAMP, C, and Pde1 belonging to the fuzzy module. As it can be noted, in the case of a very small value of Δ with respect to the scale of the phenomena modeled by the fuzzy rules (*e.g.*, $\Delta = 5$, Figure 8.12a), the system presents damped oscillations slowly converging to an attractor steady state. By using a larger value of Δ (*e.g.*, $\Delta = 10$, Figure 8.12b), the hybrid model reproduces the stable oscillations characterizing the behavior of the biochemical pathway. In this model, it can be observed that an increase in the value of Δ affects the frequency, the phase and the amplitude of the oscillations (see Figures 8.12c and 8.12d). As a matter of fact, since the firing of fuzzy rules is instantaneous, the time step must be chosen according to the temporal scale of the modeled phenomena:

- on the one hand, very small values of Δ could lead to small modifications of the values of the elements belonging to the interface by means of the mechanistic module, resulting in the activation of the same subset of fuzzy rules, with similar firing strength, at each fuzzy inference;
- on the other hand, large values of Δ imply long intervals between the fuzzy inferences, so that the impact of the fuzzy regime on the overall dynamics of the hybrid model is no longer relevant, and the simulation will be primarily driven by the mechanistic module.

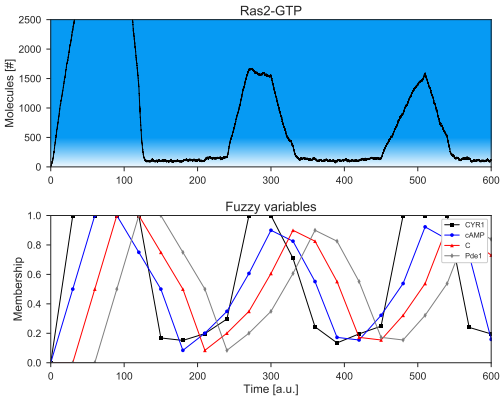


(a) $\Delta = 5$

(b) $\Delta = 10$



(c) $\Delta = 15$

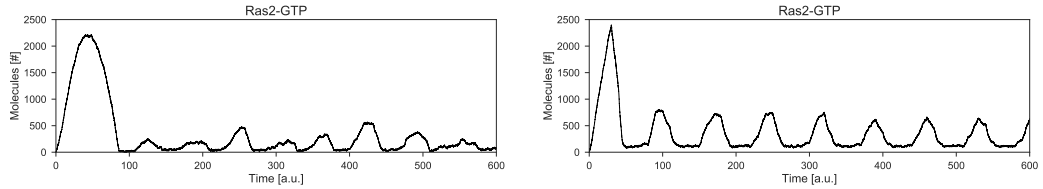


(d) $\Delta = 30$

FIGURE 8.12: Comparison of the dynamics of five key proteins in the Ras/cAMP/PKA pathway, using different settings for Δ: (a) $\Delta = 5$; (b) $\Delta = 10$; (c) $\Delta = 15$; (d) $\Delta = 30$. For each value of Δ, the dynamics of Ras2-GTP is plotted in the top plot, while the dynamics of the variables belonging to the fuzzy regime (*i.e.*, Cyr1, cAMP, C, Pde1) are plotted in the bottom plot. Ras2-GTP represents the interface between the mechanistic and fuzzy regimes; the degree of membership to the Low and High states of this protein are represented by lighter and darker shades of blue, respectively.

Therefore, Δ should be carefully chosen by the modeler, by comparing the resulting dynamics with respect to known behaviors, or considering any available data concerning the phenomena under investigation. In the case of the Ras/cAMP/PKA pathway, the model presented in [108] was exploited to produce a reference dynamics that was then used to determine the optimal value for the time step Δ (in this case, $\Delta = 10$).

Figure 8.13 shows a comparison of the dynamics of Ras2-GTP obtained by using SSA only (Figure 8.13a), and calculated by FuzzX in the hybrid model using $\Delta = 10$ (Figure 8.13b). The results show that, despite the approximations introduced by FuzzX and the intrinsic noise of the stochastic simulation, the two temporal evolutions present the same (qualitative) behavior: indeed, both simulated dynamics begin with a transient peak, in which the amount of Ras2-GTP is approximately



(a) Original model

(b) Hybrid model

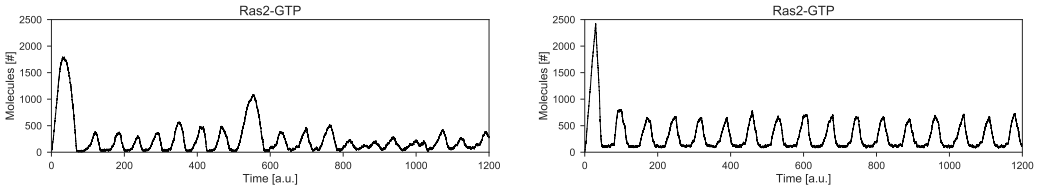
FIGURE 8.13: Comparison between the dynamics of variable Ras2-GTP in (a) the original mechanistic model [108], and in (b) the hybrid model.

equal to 2500 molecules, followed by stable oscillations. As a way to prove the accuracy of FuzzX, the mean frequency and amplitude of the oscillations obtained by simulating the original model and the hybrid model were measured and compared. Without taking into account the first transient peak, the oscillations are characterized by a mean frequency equal to $(1.17 \pm 0) \cdot 10^{-2}$ a.u. for the hybrid model, and $(1.3 \pm 0.1) \cdot 10^{-2}$ a.u. for the original model, both evaluated over 20 simulations, while the amplitude is around 500 molecules for both models (a detailed analysis of the frequency and amplitude of oscillations in the original model can be found in [207]).

These results highlight the advantages provided by FuzzX: the simplification of the model complexity by means of a human-comprehensible formalism, notably requiring a reduced number of free parameters whose values are not always available; the possibility to run a simulation of the hybrid model, which conveys both quantitative and qualitative information, exploiting a sound mathematical approach; the capability of hybrid models to accurately reproduce the outcome of a detailed mechanistic model. In addition, it is worthy of note that the computational effort of FuzzX is smaller than the execution of SSA only on the original model: a single simulation of the hybrid model showed a speed-up of approximately $25\times$, with respect to a single stochastic simulation of the original model performed by means of SSA.

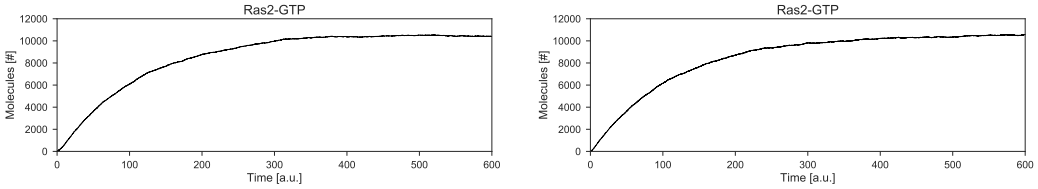
Simulations in perturbed conditions

In order to evaluate to which extent the hybrid model can reproduce known behaviors of the original model, simulations in several perturbed conditions were also performed, using $\Delta = 10$. In Figure 8.14a it is shown that Ras2-GTP exhibits a stable oscillatory dynamics in the original model, thanks to the presence of the regulative feedback loop exerted by the active unit of PKA (C) on protein Ira2. This behavior is reproduced also by the hybrid model (Figure 8.14b), where the regulative feedback loop is exerted via the fuzzy module. As a matter of fact, turning off the regulative control on Ira2 (*i.e.*, by disabling the firing of the fuzzy rules 15 and 16), the dynamics of Ras2-GTP no longer shows an oscillatory behavior (Figure 8.15b); instead, it reaches a stable steady state around $10,000$ molecules, as it was already shown in the original model (Figure 8.15a).



(a) Original model (b) Hybrid model

FIGURE 8.14: Comparison of Ras2-GTP behavior in (a) the original model, and (b) the hybrid model. The hybrid model conserves a stable oscillatory state.

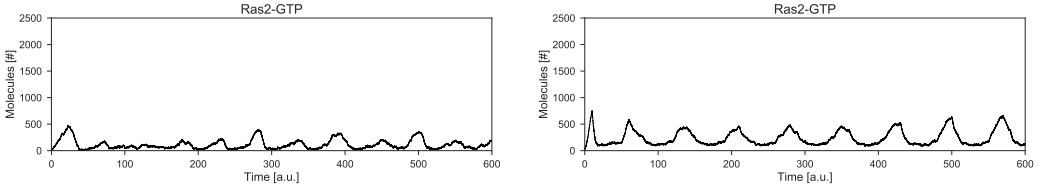


(a) Original model (b) Hybrid model

FIGURE 8.15: Comparison of Ras2-GTP behavior in (a) the original model, and (b) the hybrid model, when the feedback control on Ira2 is deactivated. Both models show the absence of oscillations and the accumulation of Ras2-GTP.

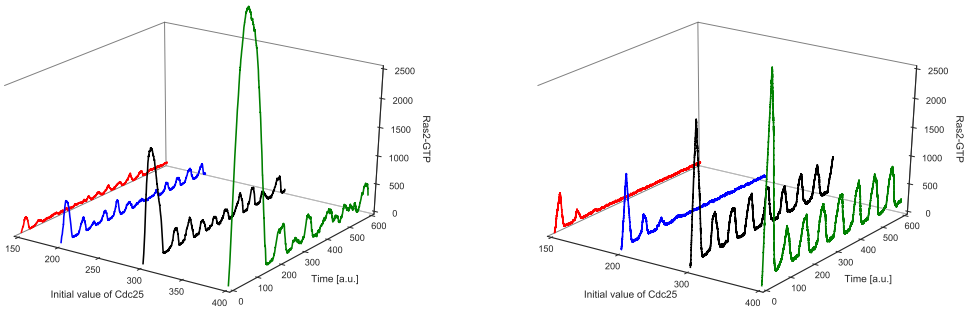
Additionally, simulations were performed, for both the original and hybrid model, starting from an initial condition that corresponds to the steady state of the system, where the regulative feedbacks are already activated. In the hybrid model, this condition is obtained by setting the initial value of Ras2-GTP to 50, cAMP to 0.9, and C to 0.9 (the values of the other variables remained unaltered with respect to Table 8.8). As it was observed in the original model [108], a stable oscillatory regime—without the initial transient peak of Ras2-GTP—is reached (Figure 8.16a). Figure 8.16b shows that the hybrid model correctly reproduces the same dynamical behavior.

Lastly, Figure 8.17 shows the dynamics obtained by varying the initial amount of Cdc25. For Cdc25 amounts lower than the reference value (*i.e.*, Cdc25 $=$ 300) the original model shows stable noisy oscillations of Ras-GTP characterized by a



(a) Original model (b) Hybrid model

FIGURE 8.16: Comparison of Ras2-GTP behavior in (a) the original model, and (b) the hybrid model, starting from a steady state condition. Both models show a stable oscillatory state without the presence of a transient phase.



(a) Original model (b) Hybrid model

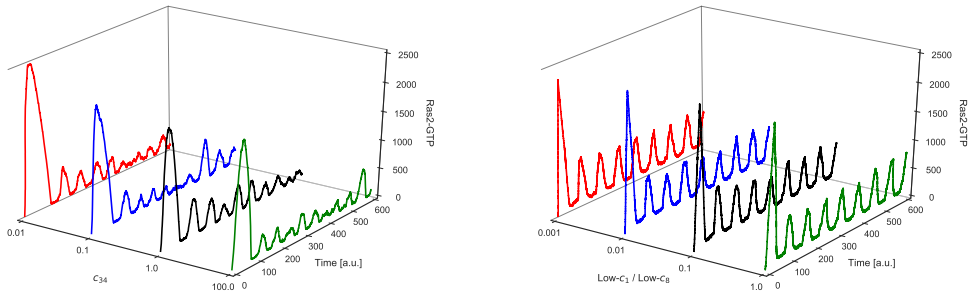
FIGURE 8.17: Perturbation of the initial value of Cdc25 in (a) the original model, and (b) the hybrid model. The dynamics obtained with the reference value (Cdc25 = 300) are depicted in black.

smaller amplitude with respect to the standard condition (Figure 8.17a), while the hybrid model (Figure 8.17b) displays damped oscillations. This behavior can be explained by the reduced effect of stochasticity in the hybrid model with respect to the original model, as it was already highlighted by the outcome of deterministic simulations of the original model [108]. Moreover, the hybrid model cannot reproduce the increased transient phase and the change in the period of oscillations, observed in the original model for higher values of Cdc25 (Figure 8.17a). This difference might be due to the approximation introduced by the choice of Δ, as it was already highlighted in Subsection 8.2.3. This issue could be bypassed by placing the feedback control—exerted by the fuzzy module—over a different variable of the mechanistic module, and not over a kinetic parameter. A thorough investigation of how the selection of different interface variables might affect the system dynamics is planned as a future work.

Analysis of fuzzy control over mechanistic parameters

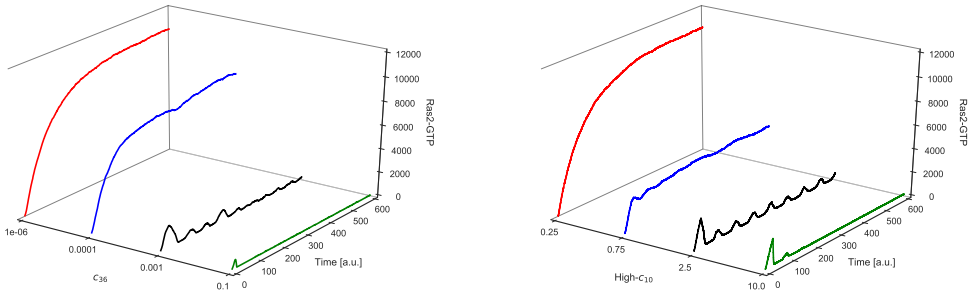
In order to assess the effect of the fuzzy control over the mechanistic parameters, simulations of the hybrid model were also performed by varying the values of the output crisp terms Low-c_1 and Low-c_8 (rule 12 and 14, respectively), which control the feedback regulation acting on Cdc25, and the value of High-c_{10} (rule 16), which controls the feedback regulation acting on Ira2. These results were compared with the analysis that was carried out on the original model in [108].

Figure 8.18b shows the dynamics of Ras2-GTP obtained by simultaneously varying the values of the terms Low-c_1 and Low-c_8. These results show that the magnitude of the feedback exerted on Cdc25 does not affect the establishment of an oscillatory regime, as it was already evidenced in the original model (Figure 8.18a). However, as opposed to the fully mechanistic representation of the biochemical pathway, the hybrid model was not able to predict the change in the frequency of oscillations that was observed in [108] by assuming a higher magnitude for the feedback regulation of C on Cdc25 (*i.e.*, higher values of Low-c_1 and Low-c_8). Again, it can be



(a) Original model (b) Hybrid model

FIGURE 8.18: Perturbation of the feedback regulation over Cdc25 in (a) the original model, and (b) the hybrid model. The dynamics were obtained by varying the parameter c_{34} in the original model, and the output crisp values Low-c_1 and Low-c_8 simultaneously in the hybrid model. The dynamics obtained with the reference values are depicted in black.



(a) Original model (b) Hybrid model

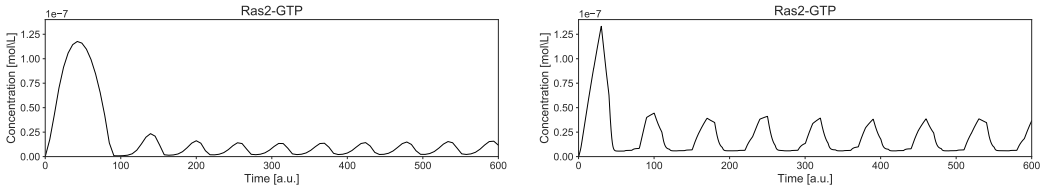
FIGURE 8.19: Perturbation of the feedback regulation over Ira2 in (a) the original model, and (b) the hybrid model. The dynamics were obtained by varying the parameter c_{36} in the original model, and the output crisp value High-c_{10} in the hybrid model. The dynamics obtained with the reference values are depicted in black.

argued that this difference might be due to the reasons discussed in Subsection 8.2.3.

Figure 8.19b reports the dynamics of Ras2-GTP obtained by varying the value of the term High-c_{10}. The results show that an increase in the magnitude of the feedback regulation exerted by Ira2 deeply affects the establishment of oscillations. Indeed, values lower than the reference one (*i.e.*, 2.5) result in the absence of oscillations and convergence to a steady state; on the other hand, higher values lead to damped oscillations or to dynamics characterized by an amount of Ras2-GTP close to zero. This behavior is in agreement with the results obtained with the mechanistic model (Figure 8.19a), as discussed in [108].

Deterministic simulations

Deterministic simulations of the hybrid Ras/cAMP/PKA model were performed to show that different formalisms can be employed to define and simulate the mechanistic module. Deterministic simulation of the hybrid model was performed by



(a) Original model

(b) Hybrid model

FIGURE 8.20: Comparison between the dynamics of variable Ras2-GTP, obtained in a deterministic simulation of (a) the original mechanistic model [108], and (b) the hybrid model.

simulating the RBM of the mechanistic module by means of the LSODA algorithm (Section 4.3). The system of ODEs was obtained by assuming MAK (Section 4.1), converting the molecular amounts into the related concentrations (in mol/L) inside a reaction volume of 30 fL (as given in [108] for *S. cerevisiae*), and transforming stochastic constants into the corresponding rate constants (see [208] for more details). Regarding LSODA functional settings, the absolute tolerance was set to 1×10^{-12} for all species, relative tolerance was set to 1×10^{-6}, and the maximum number of integration steps was set to 10000.

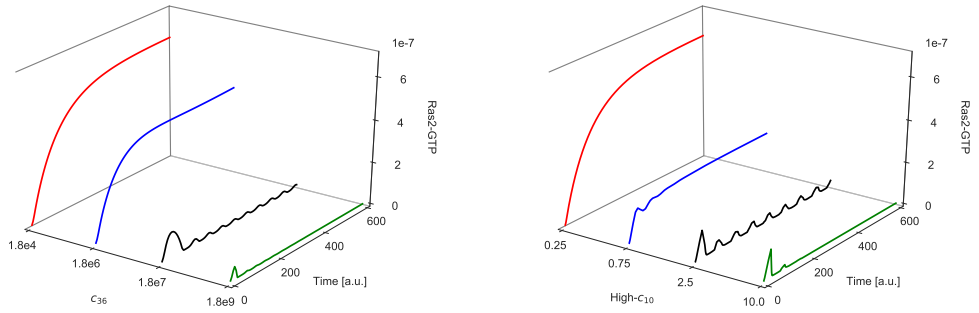
Figure 8.20 shows a comparison between the dynamics of Ras2-GTP obtained in the original model simulated with LSODA (Figure 8.20a) and in the hybrid model simulated with FuzzX (Figure 8.20b), employing the same Δ (*i.e.*, $\Delta = 10$) as for the stochastic simulations. Despite the approximations introduced with the fuzzy module, the deterministic version of the hybrid model preserves the same qualitative behavior of the fully mechanistic model, displaying dynamics characterized by the presence of a transient peak and followed by stable oscillations.

The deterministic simulation of the hybrid model displays the same emergent behavior of its stochastic counterpart also when perturbing the value of the term High-c_{10} (refer to Figure 8.19 for the stochastic simulations). Figure 8.21 shows the results concerning this analysis. As stated in Subsection 8.2.3, a change in the magnitude of the feedback regulation exerted by Ira2 affects the establishment of oscillations both in the original and in the hybrid model. In particular, decreasing the feedback regulation of Ira2 (red and blue lines in Figures 8.21a and 8.21b) results in the absence of oscillations and convergence to a steady state, while its increase leads to damped oscillations. Such behaviors are in agreement with what was already observed in [108].

8.3 Additional work in progress

8.3.1 A DFM of phenotypic transitions in stem cells

DFMs allow to explore the dynamic behavior of complex systems, even when precise mechanistic information about their components is limited. This feature of DFM can



(a) Original model (b) Hybrid model

FIGURE 8.21: Perturbation of the feedback regulation over Ira2 in deterministic simulations of (a) the original model, and (b) the hybrid model. The dynamics were obtained by varying the parameter c_{36} in the original model, and the output crisp value High-c_{10} in the hybrid model. The dynamics obtained with the reference values are depicted in black.

be exploited to model and analyze the complexity of gene regulatory networks that regulate stem cell differentiation. The study of such systems is crucial to increase both our understanding of cell fate differentiation and our ability to reprogram stem and differentiated cells for therapeutic purposes [209]. Indeed, cell differentiation is controlled by a wide and intricate network of genes, whose properties and interactions are not always fully characterized. In addition, these gene regulatory networks show many emergent behaviors, including robustness to perturbations and multiple stable states [210]. These behaviors are also due to the stochastic nature of gene expression, which must be taken into account when modeling these systems. In order to do so, a modification of the formalism presented in Chapter 5 is under development, to the aim of introducing a stochastic update of the linguistic variables to realize a stochastic DFM. As a starting point to study phenotypic transitions in stem cells, a DFM is being defined on the basis of the main components and interactions occurring in a regulatory circuit consisting in 12 genes, whose behavior was previously analyzed by means of different modeling formalisms [211, 212].

8.3.2 A hybrid model of DNA double strand breaks repair

FuzzX has been applied to analyze the repair of DNA double strand breaks in yeast cells [213], a complex biochemical system whose in-depth investigation could lead to important insights in cancer cell progression. On the one hand, this system exhibits the presence of a well-characterized process of assembly of protein complexes on the site of DNA damage, which have been modeled by means of the mechanistic module, defined as a RBM consisting in 11 species and 14 reactions; on the other hand, a complex regulative protein network, featuring several feedforward and feedback loops, is also present and it is responsible for the initiation and temporal regulation of the repair process by means of homologous recombination, a repair pathway

that exploits an intact copy of DNA to produce an error-free repair. Since quantitative information is limited, and only qualitative data is available on the regulative network, these processes have been modeled by taking advantage of a fuzzy module, corresponding to a fuzzy network that consists in 12 linguistic variables and 36 fuzzy rules. Simulations of the hybrid model can reproduce behaviors known in the literature, both in wild-type and mutant strains of *S. cerevisiae*.

Chapter 9

Conclusions and future works

Complex systems can be found everywhere in nature, ranging from physics and biology, to social or artificial systems, but for many of them a complete understanding of how complex behaviors emerge is still missing. Since the non-linear dynamic behaviors arise from negative and positive feedback regulations among a huge number of different entities, the elucidation of the mechanistic interactions that govern their functioning is an extremely challenging task from both the experimental and the computational perspectives. This is strikingly true when information about the system under investigation is limited, or it cannot be easily subject to accurate experimental measurements, or the interplay between the negative and positive control mechanisms is so complex that the system can be analyzed only by considering a subset of (usually, homogeneous) components. In order to address these problems, in the last decades scientists developed and exploited several mathematical approaches to simulate the behavior of these complex systems and grasp their inner workings, resulting in a wide variety of different formalisms.

In such context, this thesis showed that fuzzy logic can be exploited to address some of the problems that afflict complex systems modeling—namely, the lack of quantitative data, and the presence of heterogeneous components—and to bridge together different mathematical formalisms. This was achieved by presenting two main contributions, that is, DFMs and their coupling with global optimization algorithms, and FuzzX.

The DFM approach (Chapter 5 and Section 8.1) is able to address two open issues in the field of system modeling, that is, taking into account the heterogeneous nature of complex systems, and dealing with the lack of quantitative parameters that might prevent a precise characterization of some components and processes. This modeling approach may be adopted to incorporate data generated from different sources, using knowledge-based rules, and can integrate multiple data types together (*e.g.*, quantitative, semi-quantitative or qualitative, using different unit of measures). As such, DFMs not only solve some relevant issues related to the modeling and dynamical simulation of heterogeneous systems, but they are also able to provide valuable predictions that facilitate our understanding in controlling complex systems. Additionally, DFMs allow for the definition of models in a readable and simple format, through the use of linguistic variables. Their coupling with optimization algorithms

is a promising tool to uncover, in an automatic way, an optimal response of the system to an extensive number of perturbations, facilitating the design of experiments and reducing the related costs. In this thesis it was showed that this approach is able to predict the behavior ofcancer cells grown either under progressive glucose depletion or in different perturbed conditions, as well as to identify potentially novel cancer therapeutic treatments. Noteworthy, the dynamic nature and predictive power of DFMs could prove useful in assessing the effects of different types of perturbations on the behavior of the system under examination in many other application fields, such as medicine, pharmacology, etc.

Fuzzy logic can also be exploited to integrate different model formalisms and define hybrid models, able to both represent different layers of complexity (at the functional, temporal, or phenomenological level) and leverage precise mechanistic information when available [109]. The aim of this novel computational framework is to fill the gap between quantitative (mechanism-based) and qualitative models, in order to simultaneously exploit the peculiar advantages provided by each modeling approach. FuzzX (Chapter 7) was designed to model and analyze systems characterized by partial uncertainty about the underlying mechanisms, or to reduce the computational effort by mitigating the complexity of highly detailed mechanistic models. Examples of such applications are economics and finance models [214], cyber-physical systems [215], and biochemical reaction networks [27, 216].

To test its effectiveness, FuzzX was applied for the definition and analysis of a hybrid model of the Ras/cAMP/PKA pathway in yeast (Section 8.2), a biochemical system characterized by positive and negative feedback loops and molecular noise, which give rise to complex behaviors such as oscillations. According to the obtained results, FuzzX is able to qualitatively reproduce known behaviors of the system (provided that a suitable Δ is chosen), therefore proving that the interface between mechanistic approaches and fuzzy logic represents a suitable and useful framework for the modeling and simulation of complex systems. Importantly, this hybrid approach can also reduce the computational effort with respect to the analysis of the fully mechanistic model. Moreover, it was showed that different perturbations applied on variables belonging to either the mechanistic or the fuzzy module resulted in emergent behaviors that were already observed in the original model.

It is worth noting that the definition of the fuzzy module requires less precise information with respect to a fully mechanistic model: this provides a cost-effective solution (in terms of number and accuracy of parameters) to the modeling of complex systems, allowing to reproduce complex behaviors even when detailed mechanistic information is not fully known. The great flexibility of fuzzy sets allows for the definition of knowledge-driven models that can take into account the systems components that are considerably heterogeneous (*e.g.*, not representable by means of real-valued variables, and/or spanning different orders of magnitude, and/or having units of measure of different nature). In addition, fuzzy rules are human-comprehensible since they are written in simple logic statements, close to natural

language. For instance, FuzzX could facilitate the design and analysis of cyber-physical systems [215, 217], a technology that combines computational and physical components to implement real world processes, where feedback loops between the two types of components are present.

FuzzX could also be used to model and simulate systems characterized by multiple scales of time, or of spatial and functional organization, thanks to the flexibility of fuzzy sets and their ability to handle and connect qualitative and quantitative data. Moreover, FuzzX can also be applied for the revision and extension of already validated mechanistic models, in order to describe mechanisms that are not fully known; this could yield new insights on the systems under investigation, by leveraging available qualitative data and knowledge. On the other hand, whenever precise quantitative information becomes available, parts of previously defined fuzzy modules could be redefined in terms of mechanistic processes. Thanks to its semi-quantitative nature, FuzzX could be also exploited to alleviate the computational effort in black-box global optimization problems, by acting as surrogate model of a fully detailed mechanistic model during the objective function evaluations [218].

Additional improvements include devising strategies to select optimal values of Δ and developing methods for automatically inferring fuzzy rule bases from data. Indeed, even though in the application of FuzzX showed in this thesis the set of fuzzy rules was curated by an expert according to the available knowledge about the system, automatic methods based on Evolutionary Computation, such as genetic programming [219], could be coupled to FuzzX to assist in the design of the fuzzy rule base. This possibility will be investigated in the context of biochemical parameters estimation [220–222], where FuzzX will be used in the early phases of the optimization, possibly driving the candidate solutions towards regions of the search space characterized by a higher quality.

Appendix A

Fuzzy rules of the programmed cell death model

Variable	Fuzzy rules

Apoptosis	IF (Caspase3 IS Low) THEN (Apoptosis IS Low)
	IF (Caspase3 IS Medium) THEN (Apoptosis IS Medium)
	IF (Caspase3 IS High) THEN (Apoptosis IS High)

ATP	IF (DeltaPsi IS High) THEN (ATP IS High)
	IF (Glycolysis IS High) THEN (ATP IS High)
	IF (Glycolysis IS Medium) THEN (ATP IS Medium)
	IF (Glycolysis IS Low) THEN (ATP IS Low)
	IF (ComplexI IS More-functional) THEN (ATP IS High)
	IF (Glycolysis IS High) AND (DeltaPsi IS Low) THEN (ATP IS High)
	IF (Glycolysis IS High) AND (ComplexI IS Less-functional) THEN (ATP IS High)
	IF (Glycolysis IS High) AND (ComplexI IS Medium-functional) THEN (ATP IS High)
	IF (Glycolysis IS Medium) AND (DeltaPsi IS Low) THEN (ATP IS Medium)
	IF (Glycolysis IS Medium) AND (ComplexI IS Less-functional) THEN (ATP IS Medium)
	IF (Glycolysis IS Medium) AND (ComplexI IS Medium-functional) THEN (ATP IS Medium)
	IF (Glycolysis IS Low) AND (DeltaPsi IS Low) THEN (ATP IS Medium)
	IF (Glycolysis IS Low) AND (ComplexI IS Less-functional) THEN (ATP IS Very-Low)
	IF (Glycolysis IS Low) AND (ComplexI IS Medium-functional) THEN (ATP IS Medium)

Attachment	IF (N-glycosylation IS High) THEN (Attachment IS High)
	IF (N-glycosylation IS Low) THEN (Attachment IS Low)
	IF (N-glycosylation IS Medium) THEN (Attachment IS Low)

Continued on next page

Continued from previous page

Variable	Fuzzy rules

Autophagy	IF (Calcium IS High) THEN (Autophagy IS High)
	IF (BCN1 IS High) THEN (Autophagy IS High)
	IF (ROS IS High) THEN (Autophagy IS High)
	IF (Glycolysis IS Low) THEN (Autophagy is High)
	IF (PKA IS High) AND (Calcium IS Low) THEN (Autophagy IS High)
	IF (PKA IS High) AND (BCN1 IS Low) THEN (Autophagy IS High)
	IF (PKA IS High) AND (ATP IS Very-Low) THEN (Autophagy IS High)
	IF (PKA IS High) AND (ATP IS Medium) THEN (Autophagy IS High)
	IF (PKA IS High) AND (ATP IS High) THEN (Autophagy IS High)
	IF (PKA IS High) AND (Glycolysis IS High) THEN (Autophagy IS High)
	IF (PKA IS Low) AND (Calcium IS Low) THEN (Autophagy IS Low)
	IF (PKA IS Low) AND (BCN1 IS Low) THEN (Autophagy IS Low)
	IF (PKA IS Low) AND (ATP IS Very-Low) THEN (Autophagy IS Low)
	IF (PKA IS Low) AND (ATP IS Medium) THEN (Autophagy IS Low)
	IF (PKA IS Low) AND (ATP IS High) THEN (Autophagy IS Low)
	IF (PKA IS Low) AND (Glycolysis IS High) THEN (Autophagy IS Low)
	IF (PKA IS High) AND (Calcium IS Medium) THEN (Autophagy IS High)
	IF (PKA IS High) AND (BCN1 IS Medium) THEN (Autophagy IS High)
	IF (PKA IS High) AND (Glycolysis IS Medium) THEN (Autophagy IS High)
	IF (PKA IS Low) AND (Calcium IS Medium) THEN (Autophagy IS Medium)
	IF (PKA IS Low) AND (BCN1 IS Medium) THEN (Autophagy IS Medium)
	IF (PKA IS Low) AND (Glycolysis IS Medium) THEN (Autophagy IS Medium)
	IF (ATP IS Low AND PKA IS Low) AND (Calcium IS Low) THEN (Autophagy IS High)
	IF (ATP IS Low AND PKA IS Low) AND (BCN1 IS Low) THEN (Autophagy IS High)
	IF (ATP IS Low AND PKA IS Low) AND (Glycolysis IS High) THEN (Autophagy IS High)
	IF (ATP IS Very-Low AND PKA IS Low) AND (Calcium IS Low) THEN (Autophagy IS Low)
	IF (ATP IS Medium AND PKA IS Low) AND (Calcium IS Low) THEN (Autophagy IS Low)
	IF (ATP IS High AND PKA IS Low) AND (Calcium IS Low) THEN (Autophagy IS Low)

Continued on next page

Continued from previous page

Variable	Fuzzy rules

Autophagy	IF (ATP IS Very-Low AND PKA IS Low) AND (BCN1 IS Low) THEN (Autophagy IS Low)
	IF (ATP IS Medium AND PKA IS Low) AND (BCN1 IS Low) THEN (Autophagy IS Low)
	IF (ATP IS High AND PKA IS Low) AND (BCN1 IS Low) THEN (Autophagy IS Low)
	IF (ATP IS Very-Low AND PKA IS Low) AND (Glycolysis IS High) THEN (Autophagy IS Low)
	IF (ATP IS Medium AND PKA IS Low) AND (Glycolysis IS High) THEN (Autophagy IS Low)
	IF (ATP IS High AND PKA IS Low) AND (Glycolysis IS High) THEN (Autophagy IS Low)
	IF (ATP IS Low AND PKA IS Low) AND (Calcium IS Medium) THEN (Autophagy IS High)
	IF (ATP IS Low AND PKA IS Low) AND (BCN1 IS Medium) THEN (Autophagy IS High)
	IF (ATP IS Low AND PKA IS Low) AND (Glycolysis IS Medium) THEN (Autophagy IS High)
	IF (ATP IS Very-Low AND PKA IS Low) AND (Calcium IS Medium) THEN (Autophagy IS Medium)
	IF (ATP IS Medium AND PKA IS Low) AND (Calcium IS Medium) THEN (Autophagy IS Medium)
	IF (ATP IS High AND PKA IS Low) AND (Calcium IS Medium) THEN (Autophagy IS Medium)
	IF (ATP IS Very-Low AND PKA IS Low) AND (BCN1 IS Medium) THEN (Autophagy IS Medium)
	IF (ATP IS Medium AND PKA IS Low) AND (BCN1 IS Medium) THEN (Autophagy IS Medium)
	IF (ATP IS High AND PKA IS Low) AND (BCN1 IS Medium) THEN (Autophagy IS Medium)
	IF (ATP IS Very-Low AND PKA IS Low) AND (Glycolysis IS Medium) THEN (Autophagy IS Medium)
	IF (ATP IS Medium AND PKA IS Low) AND (Glycolysis IS Medium) THEN (Autophagy IS Medium)
	IF (ATP IS High AND PKA IS Low) AND (Glycolysis IS Medium) THEN (Autophagy IS Medium)
	IF (PKA IS High) THEN (Autophagy IS High)
	IF (ROS IS Medium) AND (PKA IS Low) THEN (Autophagy IS Low)

Continued on next page

Continued from previous page

Variable	Fuzzy rules
Autophagy	IF (ROS IS Medium) AND (PKA IS High) THEN (Autophagy IS Medium)
	IF (ROS IS Low) AND (PKA IS Low) THEN (Autophagy IS Low)
	IF (ROS IS Low) AND (PKA IS High) THEN (Autophagy IS High)
	IF (PKA IS High AND ATP IS Low) THEN (Autophagy IS High)
	IF (PKA IS Low AND ATP IS Low) THEN (Autophagy IS Medium)
	IF (PKA IS Low AND Calcium IS High) THEN (Autophagy IS High)
	IF (PKA IS Low AND BCN1 IS High) THEN (Autophagy IS High)
	IF (PKA IS Low AND Glycolysis IS Low) THEN (Autophagy IS High)
	IF (BCN1 IS VeryHigh) THEN (Autophagy IS High)
Bcl2	IF (CHOP IS High) THEN (Bcl2 IS Low)
	IF (CHOP IS Medium) THEN (Bcl2 IS Medium)
	IF (CHOP IS Low) THEN (Bcl2 IS High)
	IF (CHOP IS Medium) AND (JNK IS High) THEN (Bcl2 IS Low)
	IF (CHOP IS Low) AND (JNK IS High) THEN (Bcl2 IS Low)
	IF (CHOP IS Medium) AND (JNK IS Low) THEN (Bcl2 IS Medium)
	IF (BCN1 IS High) THEN (Bcl2 IS Low)
	IF (ATP IS High) THEN (Bcl2 IS High)
	IF (ATP IS Medium) THEN (Bcl2 IS High)
	IF (ATP IS Low) THEN (Bcl2 IS Low)
	IF (ATP IS Very-Low) THEN (Bcl2 IS Low)
	IF (CHOP IS High) AND (JNK IS High) THEN (Bcl2 IS Low)
	IF (JNK IS Low) THEN (Bcl2 IS High)
	IF (BCN1 IS Medium) THEN (Bcl2 IS Medium)
	IF (BCN1 IS Low) THEN (Bcl2 IS High)
	IF (BCN1 IS VeryHigh) THEN (Bcl2 IS Low)
	IF (PKA IS High) THEN (Bcl2 IS Medium)
	IF (PKA IS Low) THEN (Bcl2 IS High)
BCN1	IF (Bcl2 IS Low) THEN (BCN1 IS High)
	IF (Bcl2 IS High) THEN (BCN1 IS Low)
	IF (Bcl2 IS Medium) THEN (BCN1 IS Medium)
	IF (DAPK IS High) THEN (BCN1 IS VeryHigh)
	IF (DAPK IS Low) THEN (BCN1 IS Low)
Calcium	IF (UPR IS Low) THEN (Calcium IS Low)
	IF (UPR IS Medium) THEN (Calcium IS Medium)
	IF (UPR IS High) THEN (Calcium IS High)

Continued on next page

Continued from previous page

Variable	Fuzzy rules
Caspase3	IF (ATP IS Low) THEN (Caspase3 IS High)
	IF (ATP IS High) THEN (Caspase3 IS Low)
	IF (ATP IS Medium) THEN (Caspase3 IS Low)
	IF (ROS IS Low) THEN (Caspase3 IS Low)
	IF (ROS IS Medium) THEN (Caspase3 IS Low)
	IF (Calcium IS Medium) THEN (Caspase3 IS Low)
	IF (Calcium IS Low) THEN (Caspase3 IS Low)
	IF (ERK IS High) THEN (Caspase3 IS Low)
	IF (Bcl2 IS High) THEN (Caspase3 IS Low)
	IF (DeltaPsi IS High) THEN (Caspase3 IS Low)
	IF (ATP IS Very-Low) AND (ROS IS High) THEN (Caspase3 is Low)
	IF (ATP IS Very-Low) AND (Calcium IS High) THEN (Caspase3 is Low)
	IF (ATP IS Very-Low) AND (Bcl2 IS Low) THEN (Caspase3 is Low)
	IF (ATP IS Very-Low) AND (DeltaPsi IS Low) THEN (Caspase3 is Low)
	IF (ATP IS Low) AND (ROS IS High) THEN (Caspase3 is High)
	IF (ATP IS Medium) AND (ROS IS High) THEN (Caspase3 is High)
	IF (ATP IS High) AND (ROS IS High) THEN (Caspase3 is High)
	IF (ATP IS Low) AND (Calcium IS High) THEN (Caspase3 is High)
	IF (ATP IS Medium) AND (Calcium IS High) THEN (Caspase3 is High)
	IF (ATP IS High) AND (Calcium IS High) THEN (Caspase3 is High)
	IF (ATP IS Low) AND (Bcl2 IS Low) THEN (Caspase3 is High)
	IF (ATP IS Medium) AND (Bcl2 IS Low) THEN (Caspase3 is High)
	IF (ATP IS High) AND (Bcl2 IS Low) THEN (Caspase3 is High)
	IF (ATP IS Low) AND (DeltaPsi IS Low) THEN (Caspase3 is High)
	IF (ATP IS Medium) AND (DeltaPsi IS Low) THEN (Caspase3 is High)
	IF (ATP IS High) AND (DeltaPsi IS Low) THEN (Caspase3 is High)
	IF (ATP IS Very-Low) AND (Bcl2 IS Medium) THEN (Caspase3 is Low)
	IF (ATP IS Low) AND (Bcl2 IS Medium) THEN (Caspase3 is Medium)
	IF (ATP IS Medium) AND (Bcl2 IS Medium) THEN (Caspase3 is Medium)
	IF (ATP IS High) AND (Bcl2 IS Medium) THEN (Caspase3 is Medium)
	IF (ATP IS Very-Low) THEN (Caspase3 is Low)
	IF (ERK IS Low) THEN (Caspase3 IS Medium)
CHOP	IF (UPR IS Low) THEN (CHOP IS Low)
	IF (UPR IS Medium) THEN (CHOP IS Medium)
	IF (UPR IS High) THEN (CHOP IS High)

Continued on next page

Continued from previous page

Variable	Fuzzy rules

ComplexI	IF (PKA IS Low) THEN (ComplexI IS Medium-functional)
	IF (PKA IS High) AND (Glycolysis IS High) THEN (ComplexI IS More-functional)
	IF (PKA IS High) AND (Glycolysis IS Medium) THEN (ComplexI IS More-functional)
	IF (PKA IS Low) AND (Glycolysis IS High) THEN (ComplexI IS Medium-functional)
	IF (PKA IS Low) AND (Glycolysis IS Medium) THEN (ComplexI IS Medium-functional)
	IF (PKA IS High) AND (Glycolysis IS Low) THEN (ComplexI IS More-functional)
	IF (PKA IS Low) AND (Glycolysis IS Low) THEN (ComplexI IS Less-functional)

DAPK	IF (SRC IS High) THEN (DAPK IS Low)
	IF (Calcium IS Low) THEN (DAPK IS Low)
	IF (ERK IS High) THEN (DAPK IS Low)
	IF (SRC IS Low) THEN (DAPK IS High)
	IF (Calcium IS High) THEN (DAPK IS High)
	IF (Calcium IS Medium) THEN (DAPK IS High)
	IF (ERK IS Low) THEN (DAPK IS High)

DeltaPsi	IF (Glycolysis IS Medium) THEN (DeltaPsi IS High)
	IF (Glycolysis IS Low) THEN (DeltaPsi IS Low)
	IF (ComplexI IS More-functional) THEN (DeltaPsi IS High)
	IF (ComplexI IS Medium-functional) THEN (DeltaPsi IS High)
	IF (Calcium IS Low) THEN (DeltaPsi IS High)
	IF (Bcl2 IS High) THEN (DeltaPsi IS High)
	IF (Glycolysis IS High) AND (ComplexI IS Less-functional) THEN (DeltaPsi IS High)
	IF (Glycolysis IS High) AND (Calcium IS High) THEN (DeltaPsi IS High)
	IF (Glycolysis IS High) AND (Calcium IS Medium) THEN (DeltaPsi IS High)
	IF (Glycolysis IS High) AND (Bcl2 IS Low) THEN (DeltaPsi IS High)
	IF (Glycolysis IS Medium) AND (ComplexI IS Less-functional) THEN (DeltaPsi IS Low)
	IF (Glycolysis IS Low) AND (ComplexI IS Less-functional) THEN (DeltaPsi IS Low)
	IF (Glycolysis IS Medium) AND (Calcium IS High) THEN (DeltaPsi IS Low)

Continued on next page

Continued from previous page

Variable	Fuzzy rules

DeltaPsi	IF (Glycolysis IS Medium) AND (Calcium IS Medium) THEN (DeltaPsi IS Low)
	IF (Glycolysis IS Low) AND (Calcium IS High) THEN (DeltaPsi IS Low)
	IF (Glycolysis IS Low) AND (Calcium IS Medium) THEN (DeltaPsi IS Low)
	IF (Glycolysis IS Medium) AND (Bcl2 IS Low) THEN (DeltaPsi IS Low)
	IF (Glycolysis IS Low) AND (Bcl2 IS Low) THEN (DeltaPsi IS Low)
	IF (Bcl2 IS Medium) THEN (DeltaPsi IS Low)
	IF (Glycolysis IS High) THEN (DeltaPsi IS High)

ERK	IF (SRC IS High) THEN (ERK IS High)
	IF (RAS-GTP IS On) AND (SRC IS Low) THEN (ERK IS High)
	IF (RAS-GTP IS On) AND (DAPK IS High) THEN (ERK IS High)
	IF (RAS-GTP IS Off) AND (SRC IS Low) THEN (ERK IS Low)
	IF (RAS-GTP IS Off) AND (DAPK IS High) THEN (ERK IS Low)
	IF (RAS-GTP IS On) THEN (ERK IS High)
	IF (DAPK IS Low) THEN (ERK IS High)
	IF (RAS-GTP IS Off) AND (SRC IS High) THEN (ERK IS High)
	IF (RAS-GTP IS Off) AND (DAPK IS Low) THEN (ERK IS High)

Glycolysis	IF (Glucose IS High) THEN (Glycolysis IS High)
	IF (Glucose IS Medium) THEN (Glycolysis IS Medium)
	IF (Glucose IS Low) THEN (Glycolysis IS Low)

HBP	IF (Glucose IS High) AND (Autophagy IS Low) THEN (HBP IS High)
	IF (Glucose IS High) AND (Autophagy IS Medium) THEN (HBP IS High)
	IF (Glucose IS High) AND (Autophagy IS High) THEN (HBP IS High)
	IF (Glucose IS Medium) AND (Autophagy IS Low) THEN (HBP IS Medium)
	IF (Glucose IS Medium) AND (Autophagy IS Medium) THEN (HBP IS Medium)
	IF (Glucose IS Medium) AND (Autophagy IS High) THEN (HBP IS High)
	IF (Glucose IS Low) AND (Autophagy IS Low) THEN (HBP IS Low)
	IF (Glucose IS Low) AND (Autophagy IS Medium) THEN (HBP IS Medium)
	IF (Glucose IS Low) AND (Autophagy IS High) THEN (HBP IS High)

JNK	IF (UPR IS High) THEN (JNK IS High)
	IF (UPR IS Medium) THEN (JNK IS High)
	IF (UPR IS Low) THEN (JNK IS Low)

Continued on next page

Continued from previous page

Variable	Fuzzy rules
Necrosis	IF (Bcl2 IS High) THEN (Necrosis IS Low)
	IF (Bcl2 IS Medium) THEN (Necrosis IS Low)
	IF (ATP IS Very-Low) THEN (Necrosis IS High)
	IF (ATP IS Low) THEN (Necrosis IS Low)
	IF (ATP IS Medium) THEN (Necrosis IS Low)
	IF (ROS IS Medium) THEN (Necrosis IS Low)
	IF (ROS IS Low) THEN (Necrosis IS Low)
	IF (ATP IS High) AND (Bcl2 IS Low) THEN (Necrosis is Low)
	IF (ATP IS High) AND (ROS IS High) THEN (Necrosis is Low)
	IF (ATP IS Very-Low) AND (Bcl2 IS Low) THEN (Necrosis is High)
	IF (ATP IS Low) AND (Bcl2 IS Low) THEN (Necrosis is High)
	IF (ATP IS Medium) AND (Bcl2 IS Low) THEN (Necrosis is High)
	IF (ATP IS Very-Low) AND (ROS IS High) THEN (Necrosis is High)
	IF (ATP IS Low) AND (ROS IS High) THEN (Necrosis is High)
	IF (ATP IS Medium) AND (ROS IS High) THEN (Necrosis is High)
	IF (ATP IS High) THEN (Necrosis is Low)
N-glyco-sylation	IF (HBP IS High) THEN (N-glycosylation IS High)
	IF (HBP IS Medium) THEN (N-glycosylation IS Medium)
	IF (HBP IS Low) THEN (N-glycosylation IS Low)
	IF (ATP IS High) THEN (N-glycosylation IS High)
	IF (ATP IS Medium) THEN (N-glycosylation IS High)
	IF (ATP IS Low) THEN (N-glycosylation IS Medium)
	IF (ATP IS VeryLow) THEN (N-glycosylation IS Low)
ROS	IF (ComplexI IS Less-functional) THEN (ROS IS High)
	IF (ComplexI IS Medium-functional) THEN (ROS IS Medium)
	IF (ComplexI IS More-functional) THEN (ROS IS Low)
	IF (DeltaPsi IS Low) THEN (ROS IS High)
	IF (DeltaPsi IS High) THEN (ROS IS Medium)
SRC	IF (Attachment IS Low) THEN (SRC IS Low)
	IF (Attachment IS High) THEN (SRC IS High)
	IF (PKA IS High) THEN (SRC IS High)
	IF (PKA IS Low) THEN (SRC IS Low)
	IF (PKA IS High) AND (Attachment IS Low) THEN (SRC IS High)

Continued on next page

Continued from previous page

Variable	Fuzzy rules
Survival	IF (DeltaPsi IS Low) THEN (Survival IS Low)
	IF (DeltaPsi IS High) THEN (Survival IS High)
	IF (Caspase3 IS High) THEN (Survival IS Low)
	IF (Caspase3 IS Low) THEN (Survival IS High)
	IF (Caspase3 IS Medium) THEN (Survival IS Low)
	IF (ATP IS Very-Low) THEN (Survival IS Low)
	IF (ATP IS Low) THEN (Survival IS Low)
	IF (ATP IS Medium) THEN (Survival IS High)
	IF (ATP IS High) THEN (Survival IS High)
	IF (Autophagy IS Medium) THEN (Survival IS High)
	IF (Autophagy IS High) THEN (Survival IS High)
	IF (Autophagy IS Low) THEN (Survival IS Low)
UPR	IF (N-glycosylation IS High) THEN (UPR IS Low)
	IF (N-glycosylation IS Medium) THEN (UPR IS Medium)
	IF (N-glycosylation IS Low) THEN (UPR IS High)
	IF (ATP IS High) THEN (UPR IS Low)
	IF (ATP IS Medium) THEN (UPR IS Low)
	IF (ATP IS Low) THEN (UPR IS Medium)
	IF (ATP IS Very-Low) THEN (UPR IS High)

Appendix B

Experimental methods

The laboratory experiments presented in this work were carried out by R. Palorini, G. Votta, F. Ricciardiello, H. De Vitto and F. Chiaradonna, Department of Biotechnology and Biosciences, University of Milano-Bicocca, Milano, Italy.

Compounds. All chemicals and inhibitors were purchased from Sigma-Aldrich, except for Thapsigargin (Vinci-Biochem, Firenze, Italy), H89 (AdipoGen Life Sciences, San Diego CA, USA) and Piericidin A (Enzo Life Sciences).

Flow cytometric analyses. All flow cytometric analyses were performed using a FACScan flow cytometer (Becton-Dickinson, Franklin Lakes, NJ, USA) with CellQuest software (Becton-Dickinson). Propidium Iodide (PI)/Annexin V-FITC staining was performed using Apoptosis assay kit from Immunological Sciences. In particular, 10×10^5 cells were collected in 50 μL of binding buffer and stained with 1 μL of Annexin V-FITC and 1 μL of PI, for 15 minutes at room temperature (RT). After the incubation, samples were diluted in an appropriate volume of binding buffer (10 mM HEPES/NaOH pH 7.4, 140 mM NaCl, 2.5 mM $CaCl_2$) and analyzed. To determine the intracellular Ca^{2+} concentration, 10×10^5 cells were suspended in HBSS, incubated at 37° C for 10 minutes with 0.4 μM Fluo-4-AM (Thermo Fisher Scientific), washed with HBSS, and thus analyzed.

Confocal microscopy. 1×10^5 cells/well were seeded onto clean glass slides (Knittel glass) lodged in six-well plates and incubated according to the described protocol. For detection of cell surface expression of N-linked glycoproteins, cells were stained with 5 μg/mL *P. vulgaris* (PHA-L), Alexa Fluor 488 conjugate lectin for 1 hour. After washing, the coverslips were fixed with 1% paraformaldehyde for 10 minutes RT, incubated with Dapi 2 mg/ml for 2 minutes at RT and mounted with Dabco antifading reagent (Sigma-Aldrich). Cells were examined under a A1R Nikon laser scanning fluorescence confocal microscope at a magnification of 63$\times$ to obtain a minimum of 5 frames per field. Collected fluorescence emission was quantified using NIS-Elements AR analysis 4.10.00 software by Nikon. To measure the fluorescence of the region of interest, centred on the cell membrane, each cell in the field was considered as a separate event whose intensity fluorescence is put in mean with

all events captured for the sample and reported in graph. Data presented are the means of results from five independent experiments performed in duplicate with a minimum of 15 spots per sample observed.

Adhesion assays. For attachment assays, after a treatment of 24 hours 1×10^5 cells/sample were washed, resuspended in DMEM 0.5% serum, seeded and allowed to adhere for 1 hour at 37° C in 12-well plates coated overnight at 4° C with 0.1% heat-denatured BSA. Non-adherent cells were then removed by gentle washing with PBS containing Ca^{2+} and Mg^{2+} (Euroclone), whereas adherent cells were trypsinized and counted.

Western blot analysis. Cells were harvested and disrupted in a buffer containing 50 mM HEPES pH 7.5, 150 mM NaCl, 1% (v/v) glycerol; 1% (w/v) Triton X-100, 1.5 mM $MgCl_2$, 5 mM EGTA, protease inhibitor cocktail (Sigma-Aldrich) and phosphatase inhibitors (Sigma-Aldrich). 10 to 30 μg of total protein were resolved by SDS-PAGE and transferred to the nitrocellulose membrane, which was incubated overnight with specific antibodies: vinculin (#sc-5573, 1:10000), Eif2α (#sc-133227, 1:1000) from Santa Cruz Biotechnology Inc.; cleaved caspase 3 (#9662, 1:500), LC3 (#2775, 1:1000) from Cell Signaling Technology Inc.; HGMB1 (antibody ab18256 from Abcam, 1:1000) was detected in the media of the cells.

Transcriptome analysis. Microarray transcriptional profiling of NIH3T3 K-ras cells grown along a time course of 72 hours in HG and LG (NCBI GEO database accession from GSM741368 to GSM741375 for NIH3T3 K-ras cells) were analyzed as described in [136, 166] in order to identify differentially expressed genes (DEGs) associated to specific cell processes. Gene expression data were analyzed by using several tools present in the DAVID bioinformatics database and literature searches.

Quantification and statistical analysis. Unless otherwise specified, all results presented as averages are shown as mean $\pm$ s.d. from three or more independent experiments. Statistical significance (*p$<$0.05; **p$<$0.01; ***p$<$0.001) was determined using a two-tailed Student's *t*-test. *P*-values < 0.05 were used as the cut-off for statistical significance.

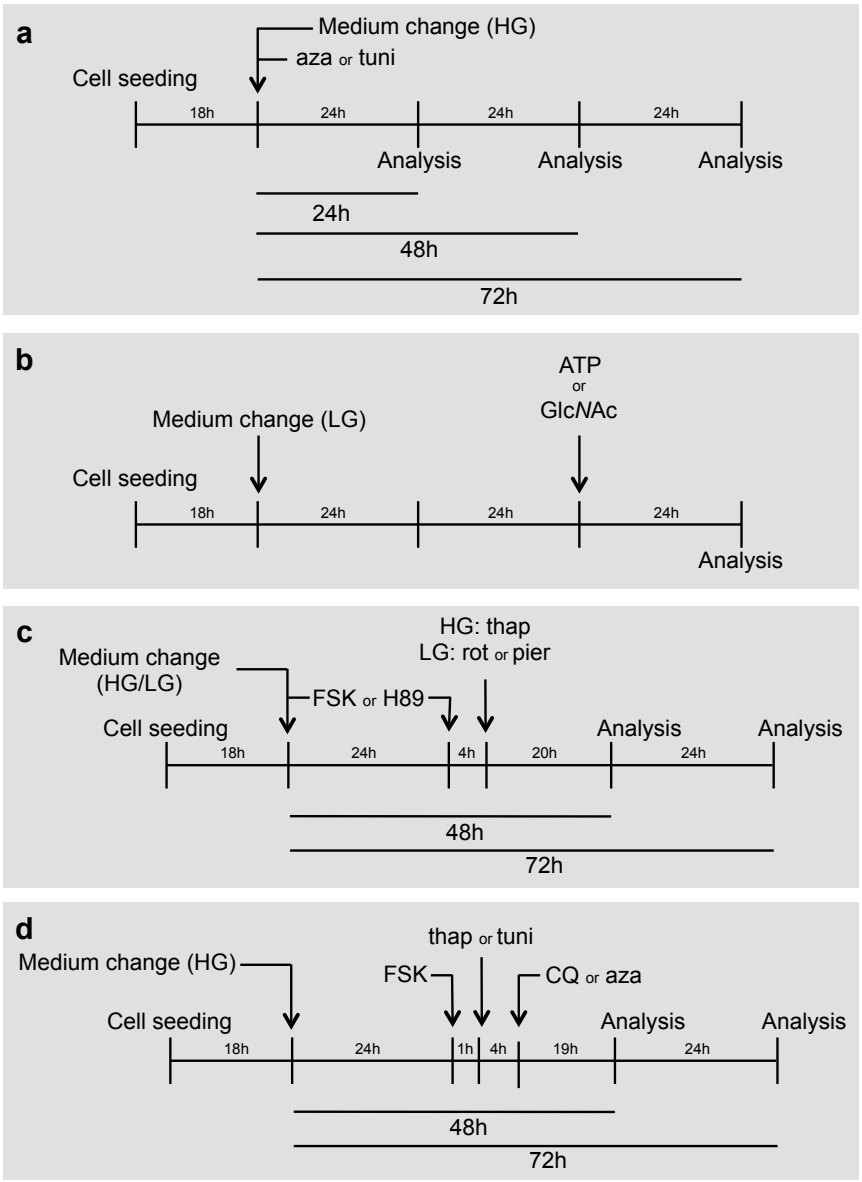


FIGURE B.1: Experimental schemes used for the biological validation of the DFM. In all experiments the medium change was performed 18 hours after cell seeding, using medium HG (25 or 5 mM glucose) or LG (1 mM glucose). Independently from the time and scheme of the treatments, the concentrations of the compounds were the following: aza 1-50 μM (1 μM in (d)), tuni 10-100 ng/mL (50 ng/mL in (d)), ATP 100 μM, Glc*N*ac 10 mM, rot 10 nM, pier 20 nM, FSK 10 μM, H89 5 μM, thap 10 nM, CQ 10 or 20 μM. Different analyses were performed at 24, 48 or 72 hours after the medium change or the first treatment, as indicated in the schemes. The viable cell count with Trypan Blue was used to calculate the percentage of cell death in the cell population.

Appendix C

Original Ras/cAMP/PKA mechanistic model

TABLE C.1: Molecular amounts of the species occurring in the initial state of the Ras/cAMP/PKA mechanistic model

Molecular species	**Copy number (molecules/cell)**
Cyr1	200
Cdc25	300
Ira2	200
Pde1	1,400
PKA	2,500
PPA2	4,000
Pde2	6,500
Ras2-GDP	20,000
GDP	$^{*}1.5 \times 10^{6}$
GTP	$^{*}5.0 \times 10^{6}$
ATP	$^{*}2.4 \times 10^{7}$

*Amounts kept constant during the execution of simulations

TABLE C.2: Reactions of the Ras/cAMP/PKA mechanistic model

	Reaction	**Stochastic constant**
P_1	Ras2-GDP + Cdc25 → Ras2-GDP-Cdc25	1.0
P_2	Ras2-GDP-Cdc25 → Ras2-GDP + Cdc25	1.0
P_3	Ras2-GDP-Cdc25 → Ras2-Cdc25 + GDP	1.5
P_4	Ras2-Cdc25 + GDP → Ras2-GDP-Cdc25	1.0
P_5	Ras2-Cdc25 + GTP → Ras2-GTP-Cdc25	1.0
P_6	Ras2-GTP-Cdc25 → Ras2-Cdc25 + GTP	1.0
P_7	Ras2-GTP-Cdc25 → Ras2-GTP + Cdc25	1.0
P_8	Ras2-GTP + Cdc25 → Ras2-GTP-Cdc25	1.0
P_9	Ras2-GTP + Ira2 → Ras2-GTP-Ira2	$^*1.0 \times 10^{-2}$
P_{10}	Ras2-GTP-Ira2 → Ras2-GDP + Ira2	$^*2.5 \times 10^{-1}$
P_{11}	Ras2-GTP + Cyr1 → Ras2-GTP-Cyr1	1.0×10^{-3}
P_{12}	Ras2-GTP-Cyr1 + ATP → Ras2-GTP-Cyr1 + cAMP	2.1×10^{-6}
P_{13}	Ras2-GTP-Cyr1 + Ira2 → Ras2-GDP + Cyr1 + Ira2	1.0×10^{-3}
P_{14}	cAMP + PKA → cAMP-PKA	1.0×10^{-5}
P_{15}	cAMP + cAMP-PKA → (2cAMP)-PKA	1.0×10^{-5}
P_{16}	cAMP + (2cAMP)-PKA →(3cAMP)-PKA	1.0×10^{-5}
P_{17}	cAMP + (3cAMP)-PKA → (4cAMP)-PKA	1.0×10^{-5}
P_{18}	(4cAMP)-PKA → cAMP + (3cAMP)-PKA	1.0×10^{-1}
P_{19}	(3cAMP)-PKA → cAMP + (2cAMP)-PKA	1.0×10^{-1}
P_{20}	(2cAMP)-PKA → cAMP + cAMP-PKA	1.0×10^{-1}
P_{21}	cAMP-PKA → cAMP + PKA	1.0×10^{-1}
P_{22}	(4cAMP)-PKA → C + C + R-2cAMP + R-2cAMP	1.0
P_{23}	R-2cAMP → R + cAMP + cAMP	1.0
P_{24}	R + C → R-C	7.5×10^{-1}
P_{25}	R-C + R-C → PKA	1.0
P_{26}	C + Pde1 → C + Pde1p	1.0×10^{-6}
P_{27}	cAMP + Pde1p →cAMP-Pde1p	1.0×10^{-1}
P_{28}	cAMP-Pde1p → cAMP + Pde1p	1.0×10^{-1}
P_{29}	cAMP-Pde1p→ AMP + Pde1p	7.5
P_{30}	Pde1p + PPA2 → Pde1 + PPA2	1.0×10^{-4}
P_{31}	cAMP + Pde2 → cAMP-Pde2	1.0×10^{-4}
P_{32}	cAMP-Pde2 → cAMP + Pde2	1.0
P_{33}	cAMP-Pde2 → AMP + Pde2	1.7
P_{34}	C + Cdc25 → C + Cdc25p	1.0
P_{35}	Cdc25p + PPA2 → Cdc25 + PPA2	1.0×10^{-2}
P_{36}	Ira2 + C → Ira2p + C	1.0×10^{-3}
P_{37}	Ras2-GTP + Ira2p → Ras2-GTP-Ira2p	1.25
P_{38}	Ras2-GTP-Ira2p → Ras2-GDP + Ira2p	2.5
P_{39}	Ira2p → Ira2	10.0

*The constants of reactions P_9 and P_{10} are changed to 3.0×10^{-2} and 7.0×10^{-1}, respectively, when reactions $P_{36}, \ldots, P_{39}$ are not active, that is, when the negative feedback of PKA on Ira2 is switched off

Bibliography

[1] Y. Bar-Yam. *Dynamics of Complex Systems*. Vol. 213. Addison-Wesley Reading, MA, 1997.

[2] A.-L. Barabási. *Linked: The New Science of Networks*. AAPT, 2003.

[3] N. Boccara. *Modeling Complex Systems*. Springer Science & Business Media, 2010.

[4] Z. Szallasi, J. Stelling, and V. Periwal. *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*. The MIT Press, 2006.

[5] G. Coyle. "Qualitative and quantitative modelling in system dynamics: some research questions". In: *System Dynamics Review: The Journal of the System Dynamics Society* 16.3 (2000), pp. 225–244.

[6] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT press, 1994.

[7] T. Eisenhammer, A. Hübler, N. Packard, and J. S. Kelso. "Modeling experimental time series with ordinary differential equations". In: *Biological Cybernetics* 65.2 (1991), pp. 107–112.

[8] B. A. Berg and A. Billoire. "Markov chain monte carlo simulations". In: *Wiley Encyclopedia of Computer Science and Engineering* (2007).

[9] C. M. Macal and M. J. North. "Tutorial on agent-based modeling and simulation". In: *Proceedings of the Winter Simulation Conference, 2005.* IEEE. 2005, 14–pp.

[10] J. Decraene and T. Hinze. "A multidisciplinary survey of computational techniques for the modelling, simulation and analysis of biochemical networks." In: *Journal of Universal Computer Science* 16.9 (2010), pp. 1152–1175.

[11] L. A. Zadeh. "Outline of a new approach to the analysis of complex systems and decision processes". In: *IEEE Transactions on Systems, Man, and Cybernetics* 1 (1973), pp. 28–44.

[12] T. Leonard and J. S. Hsu. "Bayesian methods". In: *Cambridge Series in Statistical and Probabilistic Mathematics*. Vol. 1. Cambridge University Press, 1999.

[13] F. Greil. "Boolean networks as modeling framework". In: *Frontiers in Plant Science* 3 (2012), p. 178.

[14] M. K. Morris, J. Saez-Rodriguez, P. K. Sorger, and D. A. Lauffenburger. "Logic-based models for the analysis of cell signaling networks". In: *Biochemistry* 49.15 (2010), pp. 3216–3224.

[15] H. Davulcu, M. Kifer, C. Ramakrishnan, and I. Ramakrishnan. "Logic based modeling and analysis of workflows". In: *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Vol. 98. 1998, pp. 25–33.

[16] M. Sugeno and T. Yasukawa. "A fuzzy-logic-based approach to qualitative modeling". In: *IEEE Transactions on Fuzzy Systems* 1.1 (1993), p. 7.

[17] G. Agnarsson and R. Greenlaw. *Graph Theory: Modeling, Applications, and Algorithms*. Pearson/Prentice Hall, 2007.

[18] L. A. Zadeh. "Fuzzy logic = computing with words". In: *Computing with Words in Information/Intelligent Systems 1*. Springer, 1999, pp. 3–23.

[19] L. Zadeh. "Fuzzy sets". In: *Information and Control* 8.3 (1965), pp. 338 –353. ISSN: 0019-9958.

[20] L. A. Zadeh. "Fuzzy logic". In: *Computer* 21.4 (1988), pp. 83–93.

[21] J. Yen and R. Langari. *Fuzzy Logic: Intelligence, Control, and Information*. Vol. 1. Prentice Hall Upper Saddle River, 1999.

[22] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, 1995, p. 563.

[23] J. De Kleer and J. S. Brown. "A qualitative physics based on confluences". In: *Artificial Intelligence* 24.1-3 (1984), pp. 7–83.

[24] G. Wyatt, R. Leitch, and A. Steele. "Qualitative and quantitative simulation of interacting markets". In: *Decision Support Systems* 15.2 (1995), pp. 105–113.

[25] E. Gonçalves, 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". In: *Molecular BioSystems* 9.7 (2013), pp. 1576–1583.

[26] M. Cvijovic, J. Almquist, J. Hagmar, S. Hohmann, H.-M. Kaltenbach, E. Klipp, M. Krantz, P. Mendes, S. Nelander, J. Nielsen, et al. "Bridging the gaps in systems biology". In: *Molecular Genetics and Genomics* 289.5 (2014), pp. 727–734.

[27] J. R. Karr, J. C. Sanghvi, D. N. Macklin, M. V. Gutschow, J. M. Jacobs, B. Bolival Jr, N. Assad-Garcia, J. I. Glass, and M. W. Covert. "A whole-cell computational model predicts phenotype from genotype". In: *Cell* 150.2 (2012), pp. 389–401.

[28] A. Nebot, F. E. Cellier, and M. Vallverdú. "Mixed quantitative/qualitative modeling and simulation of the cardiovascular system". In: *Computer Methods and Programs in Biomedicine* 55.2 (1998), pp. 127–155.

[29] B. Kuipers and D. Berleant. "Using Incomplete Quantitative Knowledge In Qualitative Reasoning". In: *AAAI*. Vol. 88. 1988, pp. 324–329.

[30] D. Berleant and B. J. Kuipers. "Qualitative and quantitative simulation: bridging the gap". In: *Artificial Intelligence* 95.2 (1997), pp. 215–255.

[31] K. D. Forbus and B. Falkenhainer. "Self-Explanatory Simulations: an Integration of Qualitative and Quantitative Knowledge". In: *AAAI*. Vol. 1. 1990, pp. 380–387.

[32] D. Dubois and H. Prade. "The three semantics of fuzzy sets". In: *Fuzzy Sets and Systems* 90.2 (1997), pp. 141–150.

[33] L. A. Zadeh. "Fuzzy sets as a basis for a theory of possibility". In: *Fuzzy Sets and Systems* 1.1 (1978), pp. 3–28.

[34] J. M. Mendel and R. B. John. "Type-2 fuzzy sets made simple". In: *IEEE Transactions on Fuzzy Systems* 10.2 (2002), pp. 117–127.

[35] L. A. Zadeh. "The concept of a linguistic variable and its application to approximate reasoning—I". In: *Information Sciences* 8.3 (1975), pp. 199–249.

[36] L. A. Zadeh. "The concept of a linguistic variable and its application to approximate reasoning—II". In: *Information Sciences* 8.4 (1975), pp. 301–357.

[37] L. A. Zadeh. "The concept of a linguistic variable and its application to approximate reasoning-III". In: *Information Sciences* 9.1 (1975), pp. 43–80.

[38] D. Dubois and H. Prade. "What are fuzzy rules and how to use them". In: *Fuzzy sets and systems* 84.2 (1996), pp. 169–185.

[39] V. Cherkassky. "Fuzzy inference systems: a critical review". In: *Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications*. Springer, 1998, pp. 177–197.

[40] L.-X. Wang. "Fuzzy systems are universal approximators". In: *[1992 Proceedings] IEEE International Conference on Fuzzy Systems*. IEEE. 1992, pp. 1163–1170.

[41] J. L. Castro. "Fuzzy logic controllers are universal approximators". In: *IEEE Transactions on Systems, Man, and Cybernetics* 25.4 (1995), pp. 629–635.

[42] J. L. Castro and M. Delgado. "Fuzzy systems with defuzzification are universal approximators". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26.1 (1996), pp. 149–152.

[43] E. H. Mamdani and S. Assilian. "An experiment in linguistic synthesis with a fuzzy logic controller". In: *International Journal of Man-Machine Studies* 7.1 (1975), pp. 1–13.

[44] L.-X. Wang and J. M. Mendel. "Generating fuzzy rules by learning from examples". In: *IEEE Transactions on Systems, Man, and Cybernetics* 22.6 (1992), pp. 1414–1427.

[45] T. Takagi and M. Sugeno. "Fuzzy identification of systems and its applications to modeling and control". In: *IEEE Transactions on Systems, Man, and Cybernetics* 1 (1985), pp. 116–132.

[46] M. Takács. "Critical analysis of various known methods for approximate reasoning in fuzzy logic control". In: *5th International Symposium of Hungarian Researchers on Computational Intelligence*. Hungary Budapest. 2004.

[47] L.-X. Wang. "Analysis and design of hierarchical fuzzy systems". In: *IEEE Transactions on Fuzzy Systems* 7.5 (1999), pp. 617–624.

[48] K.-Y. Tu, T.-T. Lee, and W.-J. Wang. "Design of a multi-layer fuzzy logic controller for multi-input multi-output systems". In: *Fuzzy Sets and Systems* 111.2 (2000), pp. 199–214.

[49] A. Gegov. *Fuzzy Networks for Complex Systems*. Springer, 2010.

[50] H. Kawamura. "Fuzzy network for decision support systems". In: *Fuzzy Sets and Systems* 58.1 (1993), pp. 59–72.

[51] P.-T. Chang and E. Lee. "Fuzzy decision networks and deconvolution". In: *Computers & Mathematics with Applications* 37.11-12 (1999), pp. 53–63.

[52] A. M. Yaakob, A. Serguieva, and A. Gegov. "FN-TOPSIS: fuzzy networks for ranking traded equities". In: *IEEE Transactions on Fuzzy Systems* 25.2 (2017), pp. 315–332.

[53] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 2013.

[54] Y. D. Sergeyev, D. Kvasov, and M. Mukhametzhanov. "On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget". In: *Scientific Reports* 8.1 (2018), p. 453.

[55] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. "Equation of state calculations by fast computing machines". In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092.

[56] S. Kirkpatrick, D. Gelatt, and M. Vecchi. "Optimization by simulated annealing". In: *Science* 220.4598 (1983), pp. 671–680.

[57] D. Besozzi. "Reaction-Based Models of Biochemical Networks". In: *Pursuit of the Universal. 12th Conference on Computability in Europe, CiE 2016, Proceedings*. Ed. by A. Beckmann, L. Bienvenu, and N. Jonoska. Vol. 9709. LNCS. Switzerland: Springer International Publishing, 2016, pp. 24–34.

[58] D. L. Nelson, A. L. Lehninger, and M. M. Cox. *Lehninger Principles of Biochemistry*. Macmillan, 2008.

[59] P. Érdi and J. Tóth. *Mathematical Models of Chemical Reactions: Theory and Applications of Deterministic and Stochastic Models*. Manchester University Press, 1989.

[60] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2016.

[61] M. N. Spijker. "Error propagation in Runge-Kutta methods". In: *Applied Numerical Mathematics* 22.1-3 (1996), pp. 309–325.

[62] L. Petzold. "Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations". In: *SIAM Journal on Scientific and Statistical Computing* 4.1 (1983), pp. 136–148.

[63] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. Vol. 1. Elsevier, 1992.

[64] W. S. Hlavacek, J. R. Faeder, M. L. Blinov, R. G. Posner, M. Hucka, and W. Fontana. "Rules for modeling signal-transduction systems". In: *Science Signaling* 2006.344 (2006), re6–re6.

[65] M. W. Sneddon, J. R. Faeder, and T. Emonet. "Efficient modeling, simulation and coarse-graining of biological complexity with NFsim". In: *Nature Methods* 8.2 (2011), p. 177.

[66] T. Jahnke and W. Huisinga. "Solving the Chemical Master Equation for monomolecular reaction systems analytically". In: *Journal of Mathematical Biology* 54.1 (2007), pp. 1–26.

[67] D. T. Gillespie. "Exact stochastic simulation of coupled chemical reactions". In: *The Journal of Physical Chemistry* 81.25 (1977), pp. 2340–2361.

[68] D. T. Gillespie. "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions". In: *Journal of Computational Physics* 22.4 (1976), pp. 403–434.

[69] M. A. Gibson and J. Bruck. "Efficient exact stochastic simulation of chemical systems with many species and many channels". In: *The Journal of Physical Chemistry A* 104.9 (2000), pp. 1876–1889.

[70] D. T. Gillespie. "Approximate accelerated stochastic simulation of chemically reacting systems". In: *The Journal of Chemical Physics* 115.4 (2001), pp. 1716–1733.

[71] D. T. Gillespie. "The chemical Langevin equation". In: *The Journal of Chemical Physics* 113.1 (2000), pp. 297–306.

[72] N. Le Novere and T. S. Shimizu. "STOCHSIM: modelling of stochastic biomolecular processes". In: *Bioinformatics* 17.6 (2001), pp. 575–576.

[73] B. B. Aldridge, J. Saez-Rodriguez, J. L. Muhlich, P. K. Sorger, and D. A. Lauffenburger. "Fuzzy logic analysis of kinase pathway crosstalk in TNF/EGF/insulin-induced signaling". In: *PLoS Computational Biology* 5.4 (2009), e1000340.

[74] D.-Y. Chiu and P.-J. Chen. "Applying dynamic fuzzy model in combination with support vector machine to explore stock market dynamism". In: *International Conference on Adaptive and Natural Computing Algorithms*. Springer. 2007, pp. 246–253.

[75] H. Habbi, M. Zelmat, and B. O. Bouamama. "A dynamic fuzzy model for a drum–boiler–turbine system". In: *Automatica* 39.7 (2003), pp. 1213–1219.

[76] P. P. Groumpos and C. D. Stylios. "Modelling supervisory control systems using fuzzy cognitive maps". In: *Chaos, Solitons & Fractals* 11.1-3 (2000), pp. 329–336.

[77] M. S. Nobile et al. "Fuzzy modeling and global optimization to predict novel therapeutic targets in cancer cells". In: *Bioinformatics [pending decision]* (2019).

[78] T. E. Oliphant. "Python for scientific computing". In: *Computing in Science & Engineering* 9.3 (2007), pp. 10–20.

[79] T. E. Oliphant. *A Guide to NumPy*. Vol. 1. Trelgol Publishing USA, 2006.

[80] *PyFuzzy—Python Fuzzy Package.* 2014. URL: `http://pyfuzzy.sourceforge.net/`.

[81] P. Terrence. *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Raleigh, NC, USA, 2007.

[82] S. Spolaor, M. S. Nobile, P. Cazzaniga, and D. Besozzi. "Simpful: a Simple Fuzzy Logic Library". In: *[Manuscript under preparation]* (2019).

[83] International Electrotechnical Commission (IEC). *IEC 61131-7, Programmable Controllers Part 7 - Fuzzy Control Programming*. 2000.

[84] G. Acampora, B. Di Stefano, and A. Vitiello. "IEEE 1855: The First IEEE Standard Sponsored by IEEE Computational Intelligence Society". In: *IEEE Computational Intelligence Magazine* 11.4 (2016), pp. 4–6.

[85] MathWorks. *Fuzzy Logic Toolbox—R2019a*. 2019. URL: `https://www.mathworks.com/products/fuzzy-logic.html`.

[86] J. Rada-Vilela. *The FuzzyLite Libraries for Fuzzy Logic Control*. 2018. URL: `https://fuzzylite.com/`.

[87] S. Guillaume and B. Charnomordic. "Learning interpretable fuzzy inference systems with FisPro". In: *Information Sciences* 181.20 (2011), pp. 4409–4427.

[88] C. Wagner. "Juzzy-a java based toolkit for type-2 fuzzy logic". In: *2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*. IEEE. 2013, pp. 45–52.

[89] J. M. Soto-Hidalgo, J. M. Alonso, G. Acampora, and J. Alcalá-Fdez. "JFML: a java library to design fuzzy logic systems according to the IEEE std 1855-2016". In: *IEEE Access* 6 (2018), pp. 54952–54964.

[90] I. S. Association. *IEEE Standard for Fuzzy Markup Language*. 2016. URL: `https://standards.ieee.org/`.

[91] G. Acampora. "Fuzzy markup language: A XML based language for enabling full interoperability in fuzzy systems design". In: *On the Power of Fuzzy Markup Language*. Springer, 2013, pp. 17–31.

[92] E. Jones, T. Oliphant, P. Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001. URL: `https://www.scipy.org`.

[93] Y. Tsukamoto. "An approach to fuzzy reasoning method". In: *Advances in Fuzzy Set Theory and Applications* (1979).

[94] P. Angelov and R. Yager. "Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density". In: *2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)*. IEEE. 2011, pp. 62–69.

[95] M. B. Elowitz and S. Leibler. "A synthetic oscillatory network of transcriptional regulators". In: *Nature* 403.6767 (2000), p. 335.

[96] L. Wang and R. Langari. "Complex systems modeling via fuzzy logic". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26.1 (1996), pp. 100–106.

[97] J.-S. Jang. "ANFIS: adaptive-network-based fuzzy inference system". In: *IEEE Transactions on Systems, Man, and Cybernetics* 23.3 (1993), pp. 665–685.

[98] S.-G. Cao, N. W. Rees, and G. Feng. "Analysis and design for a class of complex control systems Part I: Fuzzy modelling and identification". In: *Automatica* 33.6 (1997), pp. 1017–1028.

[99] C. D. Stylios and P. P. Groumpos. "Modeling complex systems using fuzzy cognitive maps". In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 34.1 (2004), pp. 155–162.

[100] V. G. Kaburlasos and V. Petridis. "Fuzzy lattice neurocomputing (FLN) models". In: *Neural Networks* 13.10 (2000), pp. 1145–1170.

[101] M. Graña. "Lattice computing in hybrid intelligent systems". In: *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*. IEEE. 2012, pp. 1–5.

[102] J. Bordon, M. Moškon, N. Zimic, and M. Mraz. "Fuzzy Logic as a Computational Tool for Quantitative Modelling of Biological Systems with Uncertain Kinetic Data". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 12.5 (2015), pp. 1199–1205.

[103] F. Liu, M. Heiner, and M. Yang. "Fuzzy stochastic Petri nets for modeling biological systems with uncertain kinetic parameters". In: *PLOS ONE* 11.2 (2016), e0149674.

[104] F. E. Cellier, A. Nebot, F. Mugica, and A. D. Albornoz. "Combined qualitative/quantitative simulation models of continuous-time processes using fuzzy inductive reasoning techniques". In: *International Journal of General System* 24.1-2 (1996), pp. 95–116.

[105] S. Schivo, J. Scholma, P. E. van der Vet, M. Karperien, J. N. Post, J. van de Pol, and R. Langerak. "Modelling with ANIMO: between fuzzy logic and differential equations". In: *BMC Systems Biology* 10.1 (2016), p. 56.

[106] L. Mendoza and I. Xenarios. "A method for the generation of standardized qualitative dynamical systems of regulatory networks". In: *Theoretical Biology and Medical Modelling* 3.1 (2006), p. 13.

[107] J. Krumsiek, S. Pölsterl, D. M. Wittmann, and F. J. Theis. "Odefy - From discrete to continuous models". In: *BMC Bioinformatics* 11.1 (2010), p. 233.

[108] D. Besozzi, P. Cazzaniga, D. Pescini, G. Mauri, S. Colombo, and E. Martegani. "The role of feedback control mechanisms on the establishment of oscillatory regimes in the Ras/cAMP/PKA pathway in *S. cerevisiae*". In: *EURASIP Journal on Bioinformatics and Systems Biology* 2012.1 (2012), p. 10.

[109] S. Spolaor, M. S. Nobile, G. Mauri, P. Cazzaniga, and D. Besozzi. "Coupling Mechanistic Approaches and Fuzzy Logic to Model and Simulate Complex Systems". In: *IEEE Transactions on Fuzzy Systems* (2019).

[110] S. Spolaor. "Bridging qualitative and quantitative modeling with FuzzX". In: *20th Italian Conference on Theoretical Computer Science (ICTCS2019)*. 2019.

[111] M. G. Vander Heiden, L. C. Cantley, and C. B. Thompson. "Understanding the Warburg effect: the metabolic requirements of cell proliferation". In: *Science* 324.5930 (2009), pp. 1029–1033.

[112] R. Palorini et al. "Protein Kinase A activation promotes cancer cell resistance to glucose starvation and anoikis". In: *PLoS Genetics* 12.3 (2016), e1005931.

[113] C. Huang et al. "LIMS1 promotes pancreatic cancer cell survival under oxygen-glucose deprivation conditions by enhancing HIF1A protein translation". In: *Clinical Cancer Research* (2019). Published OnlineFirst.

[114] K. Zaugg et al. "Carnitine palmitoyltransferase 1C promotes cell survival and tumor growth under conditions of metabolic stress". In: *Genes & Development* 25.10 (2011), pp. 1041–1051.

[115] P. Ye et al. "An mTORC1-Mdm2-Drosha axis for miRNA biogenesis in response to glucose- and amino acid-deprivation". In: *Molecular Cell* 57.4 (2015), pp. 708–720.

[116] H. Endo, S. Owada, Y. Inagaki, Y. Shida, and M. Tatemichi. "Glucose starvation induces LKB1-AMPK-mediated MMP-9 expression in cancer cells". In: *Scientific Reports* 8 (2018), p. 10122.

[117] Y. Zhao, E. B. Butler, and M. Tan. "Targeting cellular metabolism to improve cancer therapeutics". In: *Cell Death & Disease* 4 (2013), e532.

[118] A. Viale, P. Pettazzoni, C. A. Lyssiotis, H. Ying, N. Sánchez, M. Marchesini, A. Carugo, T. Green, S. Seth, V. Giuliani, et al. "Oncogene ablation-resistant pancreatic cancer cells depend on mitochondrial function". In: *Nature* 514.7524 (2014), pp. 628–632.

[119] M. Elgendy, M. Cirò, A. Hosseini, J. Weiszmann, L. Mazzarella, E. Ferrari, R. Cazzoli, G. Curigliano, A. DeCensi, B. Bonanni, et al. "Combination of hypoglycemia and metformin impairs tumor metabolic plasticity and growth by modulating the PP2A-GSK3β-MCL-1 axis". In: *Cancer Cell* S1535-6108.19 (2019), pp. 30152–30157.

[120] S. Elmore. "Apoptosis: a review of programmed cell death". In: *Toxicologic Pathology* 35.4 (2007), pp. 495–516.

[121] R. C. Taylor, S. P. Cullen, and S. J. Martin. "Apoptosis: controlled demolition at the cellular level". In: *Nature Reviews Molecular Cell Biology* 9.3 (2008), p. 231.

[122] C. Hetz. "The unfolded protein response: controlling cell fate decisions under ER stress and beyond". In: *Nature Reviews Molecular Cell Biology* 13.2 (2012), pp. 89–102.

[123] C. Hetz and F. R. Papa. "The unfolded protein response and cell fate control". In: *Molecular Cell* 69.2 (2018), pp. 169–181.

[124] R. Palorini, D. De Rasmo, M. Gaviraghi, L. Sala Danna, A. Signorile, C. Cirulli, F. Chiaradonna, L. Alberghina, and S. Papa. "Oncogenic K-ras expression is associated with derangement of the cAMP/PKA pathway and forskolin-reversible alterations of mitochondrial dynamics and respiration". In: *Oncogene* 32.3 (2013), pp. 352–362.

[125] R. Palorini, T. Simonetto, C. Cirulli, and F. Chiaradonna. "Mitochondrial complex I inhibitors and forced oxidative phosphorylation synergize in inducing cancer cell death". In: *International Journal of Cell Biology* 2013 (2013).

[126] F. Ricciardiello, G. Votta, R. Palorini, I. Raccagni, L. Brunelli, A. Paiotta, F. Tinelli, G. D'Orazio, S. Valtorta, L. Gioia, et al. "Inhibition of the Hexosamine Biosynthetic Pathway by targeting PGM3 causes breast cancer growth arrest and apoptosis". In: *Cell Death & Disease* 9.3 (2018), p. 377.

[127] K. W. Moremen, M. Tiemeyer, and A. V. Nairn. "Vertebrate protein glycosylation: diversity, synthesis and function". In: *Nature Reviews Molecular Cell Biology* 13.7 (2012), p. 448.

[128] J. Gu, T. Isaji, Q. Xu, Y. Kariya, W. Gu, T. Fukuda, and Y. Du. "Potential roles of N-glycosylation in cell adhesion". In: *Glycoconjugate Journal* 29.8-9 (2012), pp. 599–607.

[129] M. D. Bootman, T. Chehab, G. Bultynck, J. B. Parys, and K. Rietdorf. "The regulation of autophagy by calcium signals: Do we have a consensus?" In: *Cell Calcium* (2017).

[130] F. Sun, X. Xu, X. Wang, and B. Zhang. "Regulation of autophagy by Ca^{2+}". In: *Tumor Biology* 37.12 (2016), pp. 15467–15476.

[131] E. Kania, B. Pająk, and A. Orzechowski. "Calcium homeostasis and ER stress in control of autophagy in cancer cells". In: *BioMed Research International* 2015 (2015).

[132] R. Kang, H. Zeh, M. Lotze, and D. Tang. "The Beclin 1 network regulates autophagy and apoptosis". In: *Cell Death and Differentiation* 18.4 (2011), p. 571.

[133] A. Efeyan, W. C. Comb, and D. M. Sabatini. "Nutrient-sensing mechanisms and pathways". In: *Nature* 517.7534 (2015), p. 302.

[134] L. Poillet-Perez, G. Despouy, R. Delage-Mourroux, and M. Boyer-Guittaut. "Interplay between ROS and autophagy in cancer cells, from tumor initiation to cancer therapy". In: *Redox Biology* 4 (2015), pp. 184–192.

[135] M. B. Azad, Y. Chen, and S. B. Gibson. "Regulation of autophagy by reactive oxygen species (ROS): implications for cancer progression and treatment". In: *Antioxidants & Redox Signaling* 11.4 (2009), pp. 777–790.

[136] R. Palorini, F. Cammarata, C. Balestrieri, A. Monestiroli, M. Vasso, C. Gelfi, L. Alberghina, and F. Chiaradonna. "Glucose starvation induces cell death in K-ras-transformed cells by interfering with the hexosamine biosynthesis pathway and activating the unfolded protein response". In: *Cell Death & Disease* 4.7 (2013), e732.

[137] X.-Z. Wang, M. Kuroda, J. Sok, N. Batchvarova, R. Kimmel, P. Chung, H. Zinszner, and D. Ron. "Identification of novel stress-induced genes downstream of chop". In: *The EMBO Journal* 17.13 (1998), pp. 3619–3630.

[138] M. C. Bassik, L. Scorrano, S. A. Oakes, T. Pozzan, and S. J. Korsmeyer. "Phosphorylation of BCL-2 regulates ER Ca^{2+} homeostasis and apoptosis". In: *The EMBO Journal* 23.5 (2004), pp. 1207–1216.

[139] H. Nishitoh, A. Matsuzawa, K. Tobiume, K. Saegusa, K. Takeda, K. Inoue, S. Hori, A. Kakizuka, and H. Ichijo. "ASK1 is essential for endoplasmic reticulum stress-induced neuronal cell death triggered by expanded polyglutamine repeats". In: *Genes & Development* 16.11 (2002), pp. 1345–1355.

[140] Y. Wei, S. C. Sinha, and B. Levine. "Dual role of JNK1-mediated phosphorylation of Bcl-2 in autophagy and apoptosis regulation". In: *Autophagy* 4.7 (2008), pp. 949–951.

[141] J.-P. Decuypere, J. B. Parys, and G. Bultynck. "Regulation of the autophagic bcl-2/beclin 1 interaction". In: *Cells* 1.3 (2012), pp. 284–312.

[142] R. T. Marquez and L. Xu. "Bcl-2: Beclin 1 complex: multiple, mechanisms regulating autophagy/apoptosis toggle switch". In: *American Journal of Cancer Research* 2.2 (2012), p. 214.

[143] S. Pattingre, A. Tassa, X. Qu, R. Garuti, X. H. Liang, N. Mizushima, M. Packer, M. D. Schneider, and B. Levine. "Bcl-2 antiapoptotic proteins inhibit Beclin 1-dependent autophagy". In: *Cell* 122.6 (2005), pp. 927–939.

[144] S. Song, K. N. Jacobson, K. M. McDermott, S. P. Reddy, A. E. Cress, H. Tang, S. M. Dudek, S. M. Black, J. G. Garcia, A. Makino, et al. "ATP promotes cell survival via regulation of cytosolic [Ca^{2+}] and Bcl-2/Bax ratio in lung cancer cells". In: *American Journal of Physiology-Cell Physiology* 310.2 (2015), pp. C99–C114.

[145] L. A. Smets, J. Van den Berg, D. Acton, B. Top, H. Van Rooij, and M Verwijs-Janssen. "BCL-2 expression and mitochondrial activity in leukemic cells with different sensitivity to glucocorticoid-induced apoptosis". In: *Blood* 84.5 (1994), pp. 1613–1619.

[146] R. K. Srivastava, A. R. Srivastava, S. J. Korsmeyer, M. Nesterova, Y. S. Cho-Chung, and D. L. Longo. "Involvement of microtubules in the regulation of Bcl2 phosphorylation and apoptosis through cyclic AMP-dependent protein kinase". In: *Molecular and Cellular Biology* 18.6 (1998), pp. 3509–3517.

[147] Y. Itano, A. Ito, T. Uehara, and Y. Nomura. "Regulation of Bcl-2 protein expression in human neuroblastoma SH-SY5Y cells: positive and negative effects of protein kinases C and A, respectively". In: *Journal of Neurochemistry* 67.1 (1996), pp. 131–137.

[148] E. Zalckvar, H. Berissi, L. Mizrachy, Y. Idelchuk, I. Koren, M. Eisenstein, H. Sabanay, R. Pinkas-Kramarski, and A. Kimchi. "DAP-kinase-mediated phosphorylation on the BH3 domain of beclin 1 promotes dissociation of beclin 1 from Bcl-XL and induction of autophagy". In: *EMBO Reports* 10.3 (2009), pp. 285–292.

[149] J. Krebs, L. B. Agellon, and M. Michalak. "Ca^{2+} homeostasis and endoplasmic reticulum (ER) stress: An integrated view of calcium signaling". In: *Biochemical and Biophysical Research Communications* 460.1 (2015), pp. 114–121.

[150] G. Kroemer, L. Galluzzi, and C. Brenner. "Mitochondrial membrane permeabilization in cell death". In: *Physiological Reviews* 87.1 (2007), pp. 99–163.

[151] N. Li, K. Ragheb, G. Lawler, J. Sturgis, B. Rajwa, J. A. Melendez, and J. P. Robinson. "Mitochondrial complex I inhibitor rotenone induces apoptosis through enhancing mitochondrial reactive oxygen species production". In: *Journal of Biological Chemistry* 278.10 (2003), pp. 8516–8525.

[152] P. Pihán, A. Carreras-Sureda, and C. Hetz. "BCL-2 family: integrating stress responses at the ER to control cell demise". In: *Cell Death and Differentiation* 24.9 (2017), p. 1478.

[153] L. A. Allan, N. Morrice, S. Brady, G. Magee, S. Pathak, and P. R. Clarke. "Inhibition of caspase-9 through phosphorylation at Thr 125 by ERK MAPK". In: *Nature Cell Biology* 5.7 (2003), p. 647.

[154] R. Anjum, P. P. Roux, B. A. Ballif, S. P. Gygi, and J. Blenis. "The tumor suppressor DAP kinase is a target of RSK-mediated survival signaling". In: *Current Biology* 15.19 (2005), pp. 1762–1767.

[155] W.-J. Wang, J.-C. Kuo, W. Ku, Y.-R. Lee, F.-C. Lin, Y.-L. Chang, Y.-M. Lin, C.-H. Chen, Y.-P. Huang, M.-J. Chiang, et al. "The tumor suppressor DAPK is reciprocally regulated by tyrosine kinase Src and phosphatase LAR". In: *Molecular Cell* 27.5 (2007), pp. 701–716.

[156] I. de Diego, J. Kuper, N. Bakalova, P. Kursula, and M. Wilmanns. "Molecular basis of the death-associated protein kinase–calcium/calmodulin regulator complex". In: *Science Signaling* 3.106 (2010), ra6.

[157] B. Simon, A.-S. Huart, and M. Wilmanns. "Molecular mechanisms of protein kinase regulation by calcium/calmodulin". In: *Bioorganic & Medicinal Chemistry* 23.12 (2015), pp. 2749–2760.

[158] S. Shimizu, Y. Eguchi, W. Kamiike, Y. Funahashi, A. Mignon, V. Lacronique, H. Matsuda, and Y. Tsujimoto. "Bcl-2 prevents apoptotic mitochondrial dysfunction by regulating proton flux". In: *Proceedings of the National Academy of Sciences* 95.4 (1998), pp. 1455–1459.

[159] S. Shimizu, Y. Eguchi, W. Kamiike, S. Waguri, Y. Uchiyama, H. Matsuda, and Y. Tsujimoto. "Bcl-2 blocks loss of mitochondrial membrane potential while ICE inhibitors act at a different step during inhibition of death induced by respiratory chain inhibitors." In: *Oncogene* 13.1 (1996), pp. 21–29.

[160] A. J. Kowaltowski, R. G. Cosso, C. B. Campos, and G. Fiskum. "Effect of Bcl-2 overexpression on mitochondrial structure and function". In: *Journal of Biological Chemistry* 277.45 (2002), pp. 42802–42807.

[161] L. D. Zorova, V. A. Popkov, E. Y. Plotnikov, D. N. Silachev, I. B. Pevzner, S. S. Jankauskas, V. A. Babenko, S. D. Zorov, A. V. Balakireva, M. Juhaszova, et al. "Mitochondrial membrane potential". In: *Analytical Biochemistry* 552 (2018), pp. 50–59.

[162] M. R. Duchen. "Mitochondria and calcium: from cell signalling to cell death". In: *The Journal of Physiology* 529.1 (2000), pp. 57–68.

[163] M. C. Mendoza, E. E. Er, and J. Blenis. "The Ras-ERK and PI3K-mTOR pathways: cross-talk and compensation". In: *Trends in Biochemical Sciences* 36.6 (2011), pp. 320–328.

[164] C.-H. Chen, W.-J. Wang, J.-C. Kuo, H.-C. Tsai, J.-R. Lin, Z.-F. Chang, and R.-H. Chen. "Bidirectional signals transduced by DAPK–ERK interaction promote the apoptotic effect of DAPK". In: *The EMBO Journal* 24.2 (2005), pp. 294–304.

[165] V. J. Fincham, M. James, M. C. Frame, and S. J. Winder. "Active ERK/MAP kinase is targeted to newly forming cell–matrix adhesions by integrin engagement and v-Src". In: *The EMBO Journal* 19.12 (2000), pp. 2911–2923.

[166] F. Chiaradonna, E. Sacco, R. Manzoni, M. Giorgio, M. Vanoni, and L. Alberghina. "Ras-dependent carbon metabolism and transformation in mouse fibroblasts". In: *Oncogene* 25.39 (2006), pp. 5391–5404.

[167] F. Chiaradonna, F. Ricciardiello, and R. Palorini. "The Nutrient-Sensing Hexosamine Biosynthetic Pathway as the Hub of Cancer Metabolic Rewiring." In: *Cells* 7.6 (2018).

[168] J. W. Dennis, I. R. Nabi, and M. Demetriou. "Metabolism, cell surface organization, and disease". In: *Cell* 139.7 (2009), pp. 1229–1241.

[169] W. A. Weihofen, M. Berger, H. Chen, W. Saenger, and S. Hinderlich. "Structures of human N-acetylglucosamine kinase in two complexes with N-acetylglucosamine and with ADP/glucose: insights into substrate specificity and regulation". In: *Journal of Molecular Biology* 364.3 (2006), pp. 388–399.

[170] F. Urano, X. Wang, A. Bertolotti, Y. Zhang, P. Chung, H. P. Harding, and D. Ron. "Coupling of stress in the ER to activation of JNK protein kinases by transmembrane protein kinase IRE1". In: *Science* 287.5453 (2000), pp. 664–666.

[171] N. Sasi, M. Hwang, J. Jaboin, I. Csiki, and B. Lu. "Regulated cell death pathways: new twists in modulation of BCL2 family function". In: *Molecular Cancer Therapeutics* (2009), pp. 1535–7163.

[172] Y. Eguchi, S. Shimizu, and Y. Tsujimoto. "Intracellular ATP levels determine cell death fate by apoptosis or necrosis". In: *Cancer Research* 57.10 (1997), pp. 1835–1840.

[173] Y. Tsujimoto. "Apoptosis and necrosis: intracellular ATP level as a determinant for cell death modes". In: *Cell Death and Differentiation* 4.6 (1997), p. 429.

[174] L. Kussmaul and J. Hirst. "The mechanism of superoxide production by NADH: ubiquinone oxidoreductase (complex I) from bovine heart mitochondria". In: *Proceedings of the National Academy of Sciences* 103.20 (2006), pp. 7607–7612.

[175] R. W. McLachlan, A. Kraemer, F. M. Helwani, E. M. Kovacs, and A. S. Yap. "E-cadherin adhesion activates c-Src signaling at cell–cell contacts". In: *Molecular Biology of the Cell* 18.8 (2007), pp. 3214–3223.

[176] S. Huveneers and E. H. Danen. "Adhesion signaling–crosstalk between integrins, Src and Rho". In: *Journal of Cell Science* 122.8 (2009), pp. 1059–1069.

[177] J.-C. Wu, Y.-C. Chen, C.-T. Kuo, H. W. Yu, Y.-Q. Chen, A. Chiou, and J.-C. Kuo. "Focal adhesion kinase-dependent focal adhesion recruitment of SH2 domains directs SRC into focal adhesions to regulate cell adhesion and migration". In: *Scientific Reports* 5 (2015), p. 18476.

[178] G. N. Armaiz-Pena, J. K. Allen, A. Cruz, R. L. Stone, A. M. Nick, Y. G. Lin, L. Y. Han, L. S. Mangala, G. J. Villares, P. Vivas-Mejia, et al. "Src activation by β-adrenoreceptors is a key switch for tumour metastasis". In: *Nature Communications* 4 (2013), p. 1403.

[179] A. Beristain, S. Molyneux, P. Joshi, N. Pomroy, M. Di Grappa, M. Chang, L. S. Kirschner, G. Privé, M. Pujana, and R Khokha. "PKA signaling drives mammary tumorigenesis through Src". In: *Oncogene* 34.9 (2015), p. 1160.

[180] G. Kroemer, L. Galluzzi, J. M. Vicencio, O. Kepp, E. Tasdemir, and M. C. Maiuri. "To die or not to die: that is the autophagic question". In: *Current Molecular Medicine* 8.2 (2008), pp. 78–91.

[181] N. Cherepanova, S. Shrimal, and R. Gilmore. "N-linked glycosylation and homeostasis of the endoplasmic reticulum". In: *Current Opinion in Cell Biology* 41 (2016), pp. 57–65.

[182] C. Xu and D. T. Ng. "Glycosylation-directed quality control of protein folding". In: *Nature Reviews Molecular Cell Biology* 16.12 (2015), p. 742.

[183] I. Braakman, J. Helenius, and A. Helenius. "Role of ATP and disulphide bonds during protein folding in the endoplasmic reticulum". In: *Nature* 356.6366 (1992), p. 260.

[184] D. Gaglio, C. M. Metallo, P. A. Gameiro, K. Hiller, L. S. Danna, C. Balestrieri, L. Alberghina, G. Stephanopoulos, and F. Chiaradonna. "Oncogenic K-Ras decouples glucose and glutamine metabolism to support cancer cell growth". In: *Molecular Systems Biology* 7.1 (2011).

[185] J. Li, Q. Huang, X. Long, X. Guo, X. Sun, X. Jin, Z. Li, T. Ren, P. Yuan, X. Huang, et al. "Mitochondrial elongation-mediated glucose metabolism reprogramming is essential for tumour cell survival during energy stress". In: *Oncogene* 36.34 (2017), p. 4901.

[186] Y. Cheng, X.-H. Gao, X.-J. Li, Q.-H. Cao, D.-D. Zhao, J.-R. Zhou, H.-X. Wu, Y. Wang, L.-J. You, H.-B. Yang, et al. "Depression promotes prostate cancer invasion and metastasis via a sympathetic-cAMP-FAK signaling pathway". In: *Oncogene* 37.22 (2018), p. 2953.

[187] S. Bonnet, S. L. Archer, J. Allalunis-Turner, A. Haromy, C. Beaulieu, R. Thompson, C. T. Lee, G. D. Lopaschuk, L. Puttagunta, S. Bonnet, et al. "A mitochondria-K^+ channel axis is suppressed in cancer and its normalization promotes apoptosis and inhibits cancer growth". In: *Cancer Cell* 11.1 (2007), pp. 37–51.

[188] K. M. Heiskanen, M. B. Bhat, H.-W. Wang, J. Ma, and A.-L. Nieminen. "Mitochondrial Depolarization Accompanies Cytochrome c Release During Apoptosis in PC6 Cells". In: *Journal of Biological Chemistry* 274.9 (1999), pp. 5654–5658.

[189] S. Wen, D. Zhu, and P. Huang. "Targeting cancer cell mitochondria as a therapeutic approach". In: *Future Medicinal Chemistry* 5.1 (2013), pp. 53–67.

[190] H.-Y. Liao, C.-M. Kao, C.-L. Yao, P.-W. Chiu, C.-C. Yao, and S.-C. Chen. "2, 4, 6-trinitrotoluene induces apoptosis via ROS-regulated mitochondrial dysfunction and endoplasmic reticulum stress in HepG2 and Hep3B cells". In: *Scientific Reports* 7.1 (2017), p. 8148.

[191] G. J. Szebeni, Á. Balázs, I. Madarász, G. Pócz, F. Ayaydin, I. Kanizsai, R. Fajka-Boja, R. Alföldi, L. Hackler Jr, and L. G. Puskás. "Achiral mannich-base curcumin analogs induce unfolded protein response and mitochondrial membrane depolarization in PANC-1 cells". In: *International Journal of Molecular Sciences* 18.10 (2017), p. 2105.

[192] M. MacFarlane, G. L. Robinson, and K. Cain. "Glucose – a sweet way to die: metabolic switching modulates tumor cell death". In: *Cell Cycle* 11.21 (2012), pp. 3919–3925.

[193] J. R. Cubillos-Ruiz, S. E. Bettigole, and L. H. Glimcher. "Tumorigenic and immunosuppressive effects of endoplasmic reticulum stress in cancer". In: *Cell* 168.4 (2017), pp. 692–706.

[194] L. Galluzzi, O. Kepp, M. G. Vander Heiden, and G. Kroemer. "Metabolic targets for cancer therapy". In: *Nature reviews Drug discovery* 12.11 (2013), p. 829.

[195] T. Chijiwa, A. Mishima, M. Hagiwara, M. Sano, K. Hayashi, T. Inoue, K. Naito, T. Toshioka, and H Hidaka. "Inhibition of forskolin-induced neurite outgrowth and protein phosphorylation by a newly synthesized selective inhibitor of cyclic AMP-dependent protein kinase, N-[2-(p-bromocinnamylamino) ethyl]-5-isoquinolinesulfonamide (H-89), of PC12D pheochromocytoma cells." In: *Journal of Biological Chemistry* 265.9 (1990), pp. 5267–5272.

[196] M. Aguileta, D. Rojas-Rivera, V. Goossens, Y. Estornes, G. Van Isterdael, P. Vandenabeele, and M. Bertrand. "A siRNA screen reveals the prosurvival effect of protein kinase A activation in conditions of unresolved endoplasmic reticulum stress". In: *Cell death and differentiation* 23.10 (2016), p. 1670.

[197] J. Li, X. Dou, S. Li, X. Zhang, Y. Zeng, and Z. Song. "Nicotinamide ameliorates palmitate-induced ER stress in hepatocytes via cAMP/PKA/CREB pathway-dependent Sirt1 upregulation". In: *Biochimica et Biophysica Acta (BBA)-Molecular Cell Research* 1853.11 (2015), pp. 2929–2936.

[198] D. S. Lark, L. R. Reese, T. E. Ryan, M. J. Torres, C. D. Smith, C.-T. Lin, and P. D. Neufer. "Protein kinase A governs oxidative phosphorylation kinetics and oxidant emitting potential at complex I". In: *Frontiers in physiology* 6 (2015), p. 332.

[199] S. Papa, D. De Rasmo, Z. Technikova-Dobrova, D. Panelli, A. Signorile, S. Scacco, V. Petruzzella, F. Papa, G. Palmisano, A. Gnoni, et al. "Respiratory chain complex I, a main regulatory target of the cAMP/PKA pathway is defective in different human diseases". In: *FEBS letters* 586.5 (2012), pp. 568–577.

[200] J. García-Bermúdez, M. Sánchez-Aragó, B. Soldevilla, A. del Arco, C. Nuevo-Tapioles, and J. M. Cuezva. "PKA phosphorylates the ATPase inhibitory factor 1 and inactivates its capacity to bind and inhibit the mitochondrial H+-ATP synthase". In: *Cell reports* 12.12 (2015), pp. 2143–2155.

[201] G. Santangelo. "Glucose signaling in *Saccharomyces cerevisiae*". In: *Microbiology and Molecular Biology Reviews* 70 (2006), pp. 253–282.

[202] J. Thevelein. "Signal transduction in yeast". In: *Yeast* 10 (1994), pp. 1753–1790.

[203] J. Thevelein and J. de Winde. "Novel sensing mechanisms and targets for the cAMP-protein kinase A pathway in the yeast *Saccharomyces cerevisiae*". In: *Molecular Microbiology* 33 (1999), pp. 904–918.

[204] C. Garmendia-Torres, A. Goldbeter, and M. Jacquet. "Nucleocytoplasmic oscillations of the yeast transcription factor Msn2: evidence for periodic PKA activation". In: *Current Biology* 17.12 (2007), pp. 1044–1049.

[205] O. Medvedik, D. W. Lamming, K. D. Kim, and D. A. Sinclair. "MSN2 and MSN4 link calorie restriction and TOR to sirtuin-mediated lifespan extension in *Saccharomyces cerevisiae*". In: *PLOS Biology* 5.10 (2007), e261.

[206] P. Cazzaniga, D. Pescini, D. Besozzi, G. Mauri, S. Colombo, and E. Martegani. "Modeling and stochastic simulation of the Ras/cAMP/PKA pathway in the yeast *Saccharomyces cerevisiae* evidences a key regulatory function for intracellular guanine nucleotides pools". In: *Journal of Biotechnology* 133.3 (2008), pp. 377–385.

[207] D. Pescini, P. Cazzaniga, D. Besozzi, G. Mauri, L. Amigoni, S. Colombo, and E. Martegani. "Simulation of the Ras/cAMP/PKA pathway in budding yeast highlights the establishment of stable oscillatory states". In: *Biotechnology Advances* 30.1 (2012), pp. 99–107.

[208] O. Wolkenhauer, M. Ullah, W. Kolch, and K.-H. Cho. "Modeling and simulation of intracellular dynamics: choosing an appropriate framework". In: *IEEE Transactions on Nanobioscience* 3.3 (2004), pp. 200–207.

[209] T. Graf and T. Enver. "Forcing cells to change lineages". In: *Nature* 462.7273 (2009), p. 587.

[210] B. Zhang and P. G. Wolynes. "Stem cell differentiation as a many-body problem". In: *Proceedings of the National Academy of Sciences* 111.28 (2014), pp. 10185–10190.

[211] S.-J. Dunn, G. Martello, B. Yordanov, S. Emmott, and A. Smith. "Defining an essential transcription factor program for naive pluripotency". In: *Science* 344.6188 (2014), pp. 1156–1160.

[212] Y. T. Lin, P. G. Hufton, E. J. Lee, and D. A. Potoyan. "A stochastic and dynamical view of pluripotency in mouse embryonic stem cells". In: *PLoS Computational Biology* 14.2 (2018), e1006000.

[213] K. K. Khanna and S. P. Jackson. "DNA double-strand breaks: signaling, repair and the cancer connection". In: *Nature Genetics* 27.3 (2001), p. 247.

[214] C. H. Hommes. "Heterogeneous agent models in economics and finance". In: *Handbook of Computational Economics* 2 (2006), pp. 1109–1186.

[215] D. Serpanos. "The cyber-physical systems revolution". In: *Computer* 51.3 (2018), pp. 70–73.

[216] S. Spolaor, M. Gribaudo, M. Iacono, T. Kadavy, Z. K. Oplatková, G. Mauri, S. Pllana, R. Senkerik, N. Stojanovic, E. Turunen, et al. "Towards human cell simulation". In: *High-Performance Modelling and Simulation for Big Data Applications*. Springer, 2019, pp. 221–249.

[217] E. A. Lee. "Cyber physical systems: Design challenges". In: *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE. 2008, pp. 363–369.

[218] D. R. Jones, M. Schonlau, and W. J. Welch. "Efficient global optimization of expensive black-box functions". In: *Journal of Global Optimization* 13.4 (1998), pp. 455–492.

[219] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[220] M. S. Nobile, A. Tangherloni, L. Rundo, S. Spolaor, D. Besozzi, G. Mauri, and P. Cazzaniga. "Computational intelligence for parameter estimation of biochemical systems". In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2018, pp. 1–8.

[221] A. Tangherloni, S. Spolaor, P. Cazzaniga, D. Besozzi, L. Rundo, G. Mauri, and M. S. Nobile. "Biochemical parameter estimation vs. benchmark functions: A comparative study of optimization performance and representation design". In: *Applied Soft Computing* 81 (2019), p. 105494.

[222] S. Spolaor, A. Tangherloni, L. Rundo, M. S. Nobile, and P. Cazzaniga. "Reboot strategies in particle swarm optimization and their impact on parameter estimation of biochemical systems". In: *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2017 IEEE Conference on*. IEEE. 2017, pp. 1–8.