Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (31st cycle)

A diversity-aware computational framework for systems biology

Roberta Bardini

****

Supervisor
Prof. Alfredo Benso

Doctoral Examination Committee:
Prof. Ferdinando Di Cunto, Università degli Studi di Torino
Prof. Lars Rønn Olsen, Technical University of Denmark
Dr. Giovanni Petri, ISI Foundation
Prof. Diego Regruto Tomalino, Politecnico di Torino
Prof. Tarynn Witten, Virginia Commonwealth University

Politecnico di Torino
July 15, 2019

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and they do not compromise in any way the rights of third parties, including those relating to the security of personal data.

....
Roberta Bardini
Turin, July 15, 2019

To Crick

No, not Francis

He left in the beginning and stays in the end.

Contents

List of Figures

List of Tables

Chapter 1

Abstract

Systems biology poses several challenges to computational sciences. As a research method, it implies they need to deal with biological complexity emerging from multi-level, multi-scale, non-linear, quantitative and stochastic systems having dynamic hierarchies of regulative layers. As a research domain, it is very diverse, and many subdomains compose it, each one with its representational specificities. The scope of this work is to allow for actual collaboration in this context, by a computational framework for knowledge management in systems biology. The framework includes a modeling approach, mostly responding to knowledge inference requirements, and a domain-specific model description language, complementing the modeling approach by dealing with knowledge representation and exchange requirements. While a variety of computational approaches to systems biology exist, they often tackle a limited subset of the existing requirements. This PhD work aims at designing and presenting an integrated framework which responds to them comprehensively. In particular, this thesis focuses on two parts of the framework: a modeling approach and a model description language.

The proposed modeling approach targets knowledge inference and representation requirements by combining the strengths of state-of-the-art solutions. Relying on the Nets-within-nets (NWNs) formalism, it provides the definition completeness of mathematical models while allowing for direct execution like computational models. Supporting different information specifications, it allows for model composition processes like hybrid models but preserving formalism uniformity. Also, it supports hierarchy and flexible abstractions. These capabilities support the construction of multi-level and multi-context models: they represent not only different organization levels from the system but also different views over them. Thanks to this versatility it is possible to model in an explicit way the role of the spatial and process contexts respectively over the system at each level. Thanks to the fact these features are made explicit, the landscape of regulations and their dynamic hierarchy emerges during execution. The proposed approach initially develops to target complex biological processes such as ontogenesis. At first, an application example for developmental biology, the VPC specification in C. Elegans, is provided. It shows excellent flexibility in representation capabilities. Two more application examples follow: the first one targets a cultured synthetic biological system and the second one focuses on the spread of antibiotic resistance within the microbiota. A limitation is that models following the proposed approach work as knowledge bases only for researchers with a background in computer science.

To make them accessible for non-expert users as well, the Biological System Description Language (BiSDL) has a high-level syntax recapitulating the domain-specific language of experimental biologists. At the same time, it also covers the low-level formalism elements. Also, BiSDL supports modularity: a description can make use of other descriptions, representing interconnected and nested models. The expert user can build up models under the multi-level, multi-context approach using BiSDL, creating re-usable modules corresponding to biological structures and processes. They can store these modules in libraries. Non-expert users can access libraries and access the knowledge stored in existing modules, as well as re-use, customize and combine them into high-level models by merely connecting them, and tuning their parameters. A custom compiler generates NWNs models from BiSDL descriptions, and a custom simulator directly simulates them. In this way, system dynamics is accessible as well to the non-expert user.

The proposed computational framework devises a modeling approach that collects contributions from the different subdomains involved, and a high-level model description language making models accessible for the non-expert users in the field. The goal is to foster true interdisciplinarity in systems biology by creating a common playground for all the stakeholders. The resulting genuinely shared perspective should allow to ask new questions, and orient the growing

technological capabilities both on the computational and high-throughput analysis techniques fronts. Ultimately, the proposed framework wants to contribute, as an enabler, to the cultural shift from multi-disciplinarity to inter-disciplinarity in systems biology.

The framework at the moment provides a prototypical version of the complete flow from BiSDL descriptions to the simulation of NWNs models. In the future, the modeling approach should be tested for scalability, considering both a broader spectrum of intracellular mechanisms and a more significant number of cells in the system. Also, the simulator should adapt to parallel computations, so to handle more computational complexity. The framework should devise complexity reduction strategies to improve computational performances. Bioinformatic pipelines should support partly data-driven models construction processes involving not only parameter identification but also model architecture. The framework should also include model analysis routines to explore models formally. A smart user interface should embed the full flow from BiSDL descriptions to simulations allowing easy model exploration and design. This interface could also rely on a visual version of BiSDL, and simulation outcomes visualization.

Chapter 2

Introduction

2.1 Short summary

This section aims to introduce systems biology from a philosophical and methodological perspective, highlighting the different resulting shades of biological complexity. As a research method, systems biology implies they need to deal with biological complexity emerging from multi-level, multi-scale, non-linear, quantitative and stochastic systems having dynamic hierarchies of regulative layers. As a research domain, systems biology is very diverse, and many subdomains compose it, each one with its representational specificities. Different requirements emerge from each phase of the knowledge management cycle, and the proposed framework intends to respond to all of them comprehensively.

2.2 Systems biology as a research method

Systems biology is a discipline that considers biological systems as a whole, rather than as compositions of subparts. Its foundations imply several paradigm shifts in the way researchers approach scientific endeavors in biology. Figure 2.1 represents how science and technology flow into one another in systems biology.

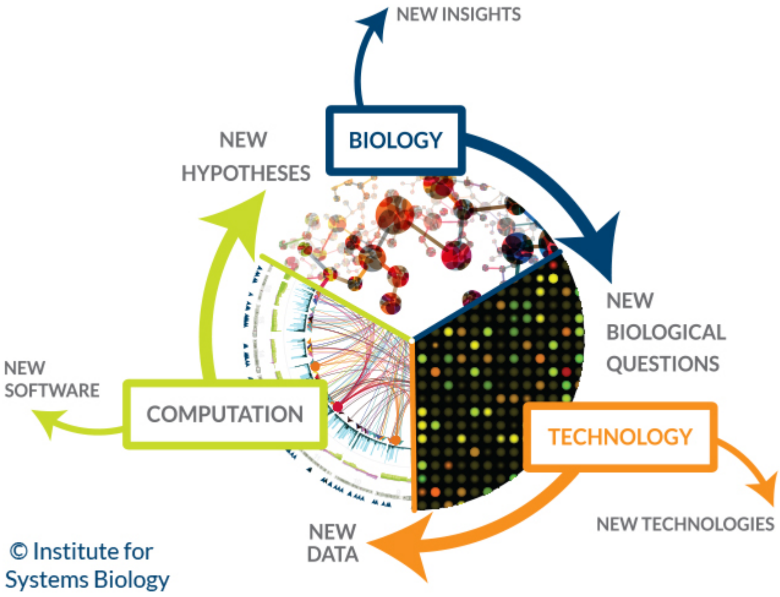


Figure 2.1. The circular pipeline of systems biology (©Institute for Systems Biology).

Starting from the sequencing of the human genome, in the last decades, technological advancements allowed for the development of high-throughput, large-scale analysis techniques of biological samples. These improvements dramatically increased the quantity and diversity of biological data made available. Extracting information from such data supported the creation of quantitative representations encompassing different system parts and organizational levels: from molecules to cells, from tissues to organisms, and from individuals to populations. On the computational side, this reflects in the creation of bioinformatic pipelines to extract information from data streams. Moreover, this allows the rise of multi-level (and multi-scale) quantitative models, comprising large amounts of the biological complexity from the system of interest, challenging representational formalisms and capabilities. Continually improving computational techniques and infrastructures allow for specifying such representations as computational models, and possibly simulate or otherwise analyze them. Computational tools generate new knowledge and insights, as well as hypotheses to be tested. Models can challenge, drive and guide experimental investigations. The capability to design experiments in a model-driven fashion optimizes the employment of powerful, high-throughput analytical techniques, which in turn allow for fine-tuned data generation to feed into bioinformatic pipelines and computational models.

2.2.1 Biology as a quantitative science

While qualitative data seek to describe a topic, quantitative data aim to quantify a phenomenon by numbers, generating structured information supporting statistical analysis. At the beginning of its history, the scientific approach to life sciences starts as qualitative. Biological behavior finds representation in functional, or sporadic annotations. The quantitative approach prevails in the collection and analysis of such annotations, rather than in the type of data gathered. Quantitative methods to data collection and interpretation in life sciences start way back in history, as told in Gregor, 2017. In the first decades of 1900, scientists begin to apply to life sciences existing tools from mathematical science. This practice later evolved using physics, information science, and engineering. For these disciplines, biology not only is an application domain gaining a benefit from the existing quantitative representation and analysis tools hard science provides. Life sciences contribute to shape and innovate these disciplines as well, posing specific challenges to them, and fostering the ideation and development of new methods and tools. By that, *hard* sciences develop not only their instruments but their theoretical foundations as well.

Computational biology can be seen as the umbrella for interminglings between *hard* and *soft* sciences. In this domain, different and complementary disciplines contribute to building up both tools to advance experimental efforts and new theoretical foundations and hypotheses to guide subsequent experimental investigations. In this context, formal and quantitative representations for current understandings of biological systems function as prediction and analysis tools, as well as knowledge bases.

Another way to intend biology as quantitative is by the large and increasing amount of data high-throughput analysis techniques make available. Not only collected data are quantitative, but they also come in vast quantities, allowing for data-driven strategies in computational biology. These combine with hypothesis-driven representations to make more of the system complexity flow into them.

2.2.2 Holism and reductionism

In Bechtel, 2017, authors describe systems biology as the territory where reductionist and holistic instances intermingle. Here, theoretical understandings of a system and mechanistic explanations of its biological functionings match forcibly. In systems biology, interestingly, this question exists at multiple system scales and levels. As highlighted in Green and Batterman, 2017, a purely reductionist approach would assume that given an ”ideal physics” and enough information from the smaller scale, it would be automatic to infer overall system behavior, encompassing all scales involved. However, biological systems are better represented by multiple organization levels rather than by different dimensional scales. Each level covers different scales and explicitly has specific boundary conditions, and finds a better representation under a different mathematical formalism. Does this make every level with its scales have an ”ideal physics” of its own? Holistic approaches consider a system as more than the sum of its parts, providing an explicit representation of the relations between them. This can provide a frame for the available (and missing) pieces of the puzzle, organizing separate, level-specific reductionist instances into an overall scheme of relations, and a higher-level context for the mechanistic explanations they produce.

2.2.3 Biology as an information science

The central dogma of molecular biology is better known as ”one gene, one protein,” meaning the simplistic hypothesis that information flows from each gene, through RNA, to a single protein, implementing a unique function. In the words of F. Crick, the central dogma rather ”*deals with the detailed residue-by-residue transfer of sequential information. It states that such information cannot be transferred back from protein to either protein or nucleic acid*”Crick, 1970. Either way, the central dogma centers on information storage supports and usage in biological systems. In the early 90s, Gilbert anticipates the next paradigm shift in life sciences W. Gilbert, 1991, a story authors tell in Lenoir, 1999. A new central dogma arises, centered on information flow instead: genetic information shapes molecular structure, which in turn implements biochemical functions, the basis of biological behavior Brutlag et al., 1994.

Biology relates to information science also in reason of the pivotal role information technology has in extracting knowledge from biological data. Oftentimes, data availability surpasses knowledge generation paces. This can be seen as an opportunity for computational approaches to improve the knowledge management cycle under many aspects. Under this perspective it is possible to distinguish computational biology and bioinformatics according to the use they make of existing capabilities from computational sciences. According to W. Gilbert, 1991, computational biology approaches the understanding of biological complexity on the theoretical front, with formal and quantitative representational tools. This includes creating *ad hoc* methods for biological problems through bioinformatics, which in the other hand functions as the experimental and instrumental side of computational biology.

2.2.4 Biology as an engineering science

As explored in Benso, Di Carlo, Politano, Savino, and Bucci, 2014, the encounter of the top-down stance of engineers and the bottom-up approach of biologists resembles a cultural clash more than a paradigm shift. When considered under an instrumental perspective, engineering practices live in their natural environment. A top-down approach benefits the design and optimization of technological means for extracting data under a fixed experimental design, including analysis, experimental pipelines, and machinery. The same holds for systematic approaches to experiments: engineering processes can optimize the application of methods and to scale it up, for example to a more significant number of samples. Standardization is another way to intend a systematic approach to life sciences. Like in the case of chemistry in the previous century, biological knowledge is undergoing a progressive, yet messy, unification and disambiguation of terms. Biology, in this case, is a challenging application domain for existing good practices and paradigms borrowed from other, pre-existing fields. In these cases, the top-down approach changes instrumental aspects of research. On the contrary, when engineering principles touch the overall scientific approach to biological subjects, the paradigm shift is structural. Historically, experimentalists lead life sciences, using bottom-up explorations. Furthermore, there is a large part of biological complexity which lacks explanations and scientific understanding. Design processes, on the other hand, rely on the implementation of function through the disposal of structure, which relies on a clear and detailed map between structure and function. In biology, the notion a structure recurrently implements a function is far from equaling a prediction tool, being a function defined in regards of the interactions a biological entity engages into at every moment as well. Top-down design processes involving biological systems must then, at the very least, accept a significant degree of uncertainty when taking a project to life.

Biotechnologies are born way earlier from the start of the genomic revolution: picking and selecting breeds on an empirical basis is, indeed, a form of genetic engineering. With a map of the genome at hand, it is possible to pursue more targeted and efficient approaches. In this field, the top-down approach actualizes by modifications of biological systems that make them useful for human purposes, and the application fields range from biomedical research to industrial production processes. Such punctual changes target the genome of an organism, determining different downstream effects. Innovation and development of genetic engineering techniques contribute to biotechnological sciences.

A more dramatic and disrupting change of perspective is that underlying synthetic biology. Rather than artificially adapting existing biological systems to a precise scope, this discipline aims to map all biological structures to their functions and to design their combinations so to implement *de novo* behavior or brand new systems reliably. In other words, synthetic biology devises the realization of living organisms following an artificial design. In this case, top-down approaches face the dramatic lack of understanding of biological systems, which impedes the definition and design of most of their parts. A (somehow ironic) demonstration of the limitations of these strategies is the fact Craig Venter creates, in 2010, an artificial cell from scratch Wade, 2010, defining a sort of minimal genetic inventory for life to subsist. This endeavor did not unveil the functional correlates of most of the genetic material employed: the artificial cell is alive, but it is not possible to explain how, exactly like in the case of non-artificial life forms.

In practice, synthetic biology often concerns the integration of genetic circuits into the genome of existing organisms, as a refined application of genetic engineering, introducing modifications that are not punctual, but rather complex. While the implementation of genetic circuits is feasible and it works on a biochemical basis *in vitro*, often synthetic biologists face the unpredictability of their biological behavior *in vivo*. This trend is an example to show how the lack of understanding of biological systems hits the intention to design their behavior. Undiscovered processes, unmapped quantitative aspects of even known mechanisms, or other unknown causes interfere with the assumptions behind the implementation. Still, if not to the scope of design and implementation, for sure synthetic biology can respond to the aim of bottom-up discovery and produce valuable data and information. It provides different experimental set-ups, helping to advance biological knowledge in general. Also, in general, synthetic designs aim at comprising more of the system complexity when extending the constructive approach of engineering to whole organisms Purnick and Weiss, 2009. Computational biology can mediate the inclusion of biological complexity into a design process, providing tools for modeling and then taking into account the context an artificial construct connects to within a biological system Bardini, Politano, et al., 2018.

2.3 Interpreting biological complexity

The previous section provided a general overview of how systems biology as a research method challenges the borders of life sciences as a research domain. This section explores how the underlying paradigm shifts reflect into methodological requirements when approaching the different shades and recurrent features of biological complexity.

2.3.1 Why complex?

Why are biological systems complex? Putting aside the age-old matter of how aliveness is more or less operationally defined, let's ask some other questions: *why does evolution tend to develop increasingly complex systems? How does hierarchical complexity emerge along evolutionary paths?* In Wolf et al., 2018, authors provide a possible theoretical explanation, stating that the emergence of complexity could base on self-organized criticalities, a concept taken from spin-glass theories. Authors claim frustrated interactions between biological actors at different system levels make complexity emerge through evolution. The reader can refer to Wolf et al., 2018 for further exploration of this concept.

This section highlights different shades of biological complexity, introducing in this way the relevant representational requirements.

2.3.2 Defining biological complexity

Living beings are complex systems. This means they are systems in the firs place, that is, it is possible to describe them as sets of entities forming a whole by means of their relationships. Such relationships can be, for example, dependencies or interactions of different sorts. Systems have boundaries which define where they end, and where the environment they live into begins.

Each part of a biological system has one or more properties (structural features) and behaviors (dynamic evolutions related to properties). The system as a whole has properties and behaviors different than the plain combination of those of its parts. This emerges in the interaction of the system with its environment.

In a complex biological system, it is not possible to directly understand which behaviors will take place given a set of properties. This can be due to an extreme sensitivity to initial conditions, emergent properties and very large numbers of parameters.

Complex biological systems often exhibit nonlinear behaviors, that is, given the same stimulus, they respond in different ways depending on the context or the previous behaviors.

Emergence can give rise to self-organization processes and spontaneous order. In these cases, a complex system in biology exhibit an emergent organization and de-centralized self-coordination.

Biological systems are adaptive as well, that is, they can retain information from past behaviors and interactions with the environment, and use that information to adapt their present behavior.

All of these aspects considered together contribute to define *biological complexity*. *Systems biology* as a research method aims at targeting biological complexity under an *holistic perspective*, that is, focusing more on the whole than on the separate parts. This reflects in the fact models not only represent biological substructures, but also the relationships between them. Also, systems biology puts together different types of biological entities, traditionally investigated in separate subdomains. This highlights multiple relevant organizational levels in biological systems, encompassing wide dimensional ranges in space and time.

There are several objectives in tackling biological complexity with models: gaining a more in-depth insight over systems functioning, representing biological entities formally and quantitatively, generating and testing hypotheses to optimize and guide experimental strategies are some of them. To make these goals achievable, it is necessary to carefully consider the modeling requirements that biological complexity with its specificities poses to systems biology models.

The first step is then to define which recurrent features of biological systems, and the perspectives under which they can be considered so to capture different shades of complexity.

2.3.3 Information dynamics

It is possible to interpret biological and biochemical interactions as a physical implementation of an information processing complex scheme. In Landauer, 1999, Landauer provides a historical perspective over the concept information is ”inevitably physical”; that is, in the material world information exists only by its physical supports. In Mehta et al., 2016, authors extend this concept, developed in the context of computational sciences, to the field of synthetic biology. The work focuses on the role of free energy usage for preserving modularity in non-equilibrium systems used as modules for synthetic biochemical circuits. Authors suggest free energy consumption in synthetic biology also happens in reason of its capability of erasing memory. They interpret biochemical modifications enacting circuit functioning as memory *writing* and *erasing* operations over biomolecules, which in this context work as information supports. The authors provide a range of biochemical implementations of memory modification operations in synthetic biology, ranging from post-translational modifications to chromatin adjustment and recombinase activity. These examples clarify how information science and physics can approach biological complexity under the information dynamics perspective.

2.3.4 Structure and function

In biology structure not only encodes and implements information and function, but it is possible to consider it as *being* the function itself. Finding structural aspects of a function is then essential. For example, functional motives in a protein act in reason of the way particular amino acids occupy a volume. The electrochemical characteristics of those amino acids contribute to defining the specific functional identity of the protein. Also, the same relation between structure and function can be explored at the protein level as well, being their biochemical profile (indicating their role as chemicals) a result of their molecular structure. Evolutionary pressure seems to act over function, causing the structure to adapt. Exploring the phylogenetic tree, it is noticeable that different structural arrangements can implement the same function: convergent evolution takes to the contemporary existence of different architectural solutions to the same functional requirements across a diversity of species.

In biology, there is a tendency to define a function in reason of interaction. In this context, interaction refers to the unilateral or mutual modification of biological structures possibly leading to structural/functional state changes. The functional annotation of biological entities stems from their attested or predicted interactions; in other words, they refer to their relative role(s) in different biological processes, and the possible interactors involved.

Functional annotation then describes one or different structures respect to their capabilities - attested or inferred - to interact with other structures, relating to what the biological structure can do in the case certain conditions are satisfied. In general, functional definition, if deprived of an interaction denotation, goes back to the mere structure. *Can an isolated DNA strand, with no transcriptional machinery attached and ready to produce transcripts out of its open reading frames, be considered to contain genes?* In a way, functions link different structures providing them with a shared context, that is, specific biological processes they can join. Interactors themselves supply the setting for the actualization of a function. The DNA strand, combined (at least) with the transcriptional machinery, makes open reading frames operationally definable as groups of genes. In another operational context, maybe the DNA would have been defined only as an acid, complex biomolecule.

Context is then pivotal for drawing which ones among all possible functional identities will come into interaction for a specific biological structure. Moreover, it is possible to define a context as a set of interactors in turn.

2.3.5 Space and time

The context points at the coexistence of the biological structures involved. Coexistence implies proximity in both space and time. If the DNA strand and the transcriptional machinery are close enough to attach, but one ceases to exist before the other appears, they will not interact. If they live at the same time, but in different locations, that is, with no possibility to physically interact, the interaction will not take place.

Then, spatiotemporal proximity transforms some of the interaction capabilities (and the respective functional annotations) of each biological structure to an actual interaction taking place in space and time.

It is possible to extend the same kind of reasoning considering a group of interactors as a single structure, to which all of the above applies. For example, the transcriptional machinery was a single interactor in the cases above, but it is possible to describe several substructures composing it. The same spatiotemporal requirements and structure/function definitions hold true for these substructures as well.

Proximity can also concern relationships, that is, it can indicate a relatedness, neighborhood relation or closeness of two elements in a generic model, for example a social network or a phylogenetic tree. In the context of biological complexity as intended in this work, spatiotemporal proximity is what builds up the model of relational schemes for the system subparts. This introduces the concept of interaction networks presented in the following paragraph.

2.3.6 Interaction networks

Representations of biological systems often focus on functional relations between elements. Networks of interactions conveniently express this. Most times, such kind of depictions abstract away from the physical system, focusing more on functional relationships than they do on structures. Since structural aspects are the ones implementing functionalities, it is necessary to take into account them in an explicit way as well. So this implies considering their existence in space dimensions, and the spatial relations intercurring between actors and the environment. Spatiality is both a feature and a mediator to regulation mechanisms. In a way, it is the substrate linking hierarchical levels and the space continuum in representations of biological systems. A good model for systems biology needs then to explicitly model spatiality.

2.3.7 Compartments and semi-permeability

A recurrent feature of living organisms is compartmentalization, that is, the sub-division of the system in, possibly concentric, subvolumes by semi-permeable membranes, providing functional segregation. In each compartment, specific processes take place. Segregating compartment-specific biological elements allows for advancing different, potentially incompatible processes in parallel, and isolating damaging biochemical reactions from the rest of the system. Each compartment specializes so that all actors relevant for a particular process are in spatial proximity and do not dilute in the overall cellular environment.

The membranes delimiting compartments are semi-permeable; that is, they selectively regulate which molecules can or can not pass through them. Exchanges of substances and signals between compartments are then strictly regulated. Selective communication by regulated permeability shows how spatiality can provide a constitutional regulation layer to the system. A good model should then, besides explicitly expressing spatiality, naturally represent both the segregation in subspaces and the selective communication mechanisms happening across the boundaries between compartments.

2.3.8 Dynamic hierarchies

The usage of genomic information is regulated at different system levels and by different mechanisms. Regulation mechanisms have dynamic hierarchical relations between them, that change according to the process context. The broader definition of epigenetic regulation is useful for delimiting this regulative scheme, devising every mechanism acting between a genotype and the corresponding possible phenotypes Hallgrimsson and Hall, 2011. The ensemble of possible regulative states for biological entities draws a complex landscape. In representing this structure, it is necessary to tune the level of abstraction in its representation so to capture its most relevant touchpoints to process evolution.

2.3.9 Complex biological processes

Systems biology aims to bring in more of the complexity from biological systems. The definitions provided so far are valid for a partial, even though systemic, screenshot of a complex biological system.

It is desirable to comprise system complexity also in reason of structural evolution along time. At this aim, it is necessary to go beyond the screenshot and watch the entire movie instead. That is, it is required to represent system organizations not only in space, and under a particular regulation set-up, but their time organization too. Comprising time organization implies to describe how the regulation set-up of the system evolves according to the process context, drawing a landscape of regulative states changing one into the other in sequential, timed ways.

Ontogenetic processes provide an excellent example of biological processes for which this approach is necessary. Ontogenesis in biology has different acceptations, and it also refers to the phenomena by which a complex organism with structural and phenotypic complexity develops out of a single cell, i. e. out of a uniform package of information Oyama, 2000. All developmental processes are ontogenetic and occur without the intervention of an external regulator. Instead, local interactions between biological actors make morphogenetic patterns emerge. Various subsequent phases compose a developmental process, each one corresponding to a different regulatory set-up.

The regulation of genomic information usage is finely tuned and context-dependent. Plus, it depends on both spatiality and timing, as in process context. Almost all cells in an organism share the same genomic information. Different phenotypes and supracellular architectures emerge.

A good model for ontogenetic processes needs to express subsequent phases, the corresponding regulatory set-ups and the passages from a state to the next. All of these elements contribute to drawing paths across a landscape of regulatory dynamical conditions. Such paths may also devise branches and rules for choosing directions. A good model needs to represent the landscape and the paths composing it, together with the regulated passages and crossroads.

Complex biological processes such as ontogenesis pose additional requirements to models for systems biology. These requirements need explicit representation, posing the challenge of specifying model parts that do not directly relate to physical structures in the system. These layers are dedicated to express the regulative landscape of the system, and of each subpart of it, explicitly.

2.4 Representing biological complexity

In systems biology, all the facets of biological complexity contribute to posing requirements to the modeling process. In addition to that, there are recurrent features in biological systems, which contribute to define the requirements posed to modeling approachs. This section explores such conditions to define the features of a good model for systems biology.

This section also aims to clarify some relevant axes over which to stretch the concept of biological complexity. Also, it seeks to identify an optimal positioning for computational models in systems biology under the presented perspective.

Multi-level and multi-scale

In biology, each interaction context can contain and belong to other interaction contexts. A context can correspond to a specific set of interacting biological structures, being part of and being composed by other structures. At the same time, it can define a bioprocess, separable into smaller processes, and composing larger ones.

Models of biological systems require efficient ways to represent context-dependent and flexible hierarchies, implying the models can express multi-level systems. Each level of organization may correspond to different dimensional ranges of interest, implying a model should support multi-scale information in both space and time dimensions.

Comprising multiple system levels brings on a systemic and holistic view considering all interconnections between subparts. For example, in a living organism, some levels of interest can be that of molecules and molecular networks, the level of cells and cellular communications, and the ones of tissues and organs respectively. An excellent computational model has then a multi-level, hierarchical architecture, and represents separately each level of interest from the system. Still, biological actors from each level coexist on a spatiotemporal continuum in the physical system, and this draws necessary interconnections between levels, which a good model should then represent in a clear and consistent way.

Multi-level and multi-scale are not synonyms: the first refers to the multiple organization levels from the system, the second to the fact biological systems and each level from them can span over extended time and space scales. The dimensional range centering over a system level does not univocally define that level, nor a level sets the boundaries for a specific scale range. Models need to express a vast range of parameter values, preserving dimensional consistency both intra-level and cross-level in the model.

Implicit and explicit

Given this, a modeler may proceed with different degrees of abstraction from the physical system, for example including in the model functional information alone, that is, making structural aspects implicit. This strategy can be useful or necessary in some cases. For example, functional information may be the only one available to build the model. It fails to capture many shades of complexity. On the other hand, the more explicitly the model treats the spatiotemporal proximity underlying interaction, the less abstraction it accepts from the physical nature of the system. Explicit modeling frees models from assumptions about the link between structure and function. Interaction between subparts happens in reason of their proximity, and may or may not occur, during simulation, given it is possible. Conditional or stochastic rules govern the actualization of interactions out of interaction capabilities. This higher-detail description corresponds to a less biased way to proceed, and it enables to express the stochastic component to biological behavior. In general, a lower degree of abstraction makes representations less coarse-grained, and it allows for showing in a clear way the interdependence of regulations and contexts. Models which follow, when possible, this direction are more suitable tools for the scopes of systems biology, both in terms of predictive power and of knowledge representation.

Qualitative and quantitative

Some biological phenomena are sometimes better described by *qualitative* information, or data availability constraints the representation to qualitative or semi-quantitative descriptions. For example, diagrammatic models, typical of many subfields of biology, represent functional information, most times covering small subparts of the system, or single bioprocesses. Such information consists, oftentimes, of categorical labels, on-off state descriptions, and absence-presence annotations.

Systems biology aims to comprise *quantative* aspects of biological systems as an essential aspect of system complexity. Gaining a more quantitative insight over the system involves the quantification of kinetic parameters of bioprocesses, as well as assessing the concentrations of biomolecules, the transcriptional rates for genes, and performing the mapping of different *omic* spaces in an high-throughput way. Quantitative information can derive from large system portions, extending the scope of models both horizontally, that is, on single levels, and vertically, including multiple system levels.

Discrete and continuous

It is possible to represent biological phenomena as composed by discrete elements interacting in separated steps, as well as by physical entities living over a continuum. Both approaches perform substantial simplifications over the system. Discrete representations divide continuous quantities into distinct parts, and subsequently easily represent encapsulation and separateness of individual biological actors. Continuous models represent the space-time continuum, under the assumption that every variable represents a compartment. This link sets the maximum resolution for that compartment, and it implies the variable sums up a supposedly uniform spatial distribution (or concentration) for the modeled quantity.

Hypothesis-driven and data-driven

Hypothesis-driven modeling tends to embed into model architectures the functional relations inferred between system parts. This approach relates to more abstract, functional representations of a system. Diagrammatic models belong to this category, but hypothesis-driven models can be quantitative as well. Data-driven modeling tends to build up model architectures and to identify parameters starting from data. As opposed to the previous approach, this aims to be completely unbiased. A combination of the two methods can provide a compromise between the necessity to embed existing knowledge in models, and the need to make them information-rich and unbiased. For example, a model can inherit its architecture from functional relations, and the value of the quantitative parameters characterizing them from data.

Deterministic and stochastic

While biological processes intrinsically exhibit stochasticity, most models in systems biology function in deterministic ways. Models should instead express non-determinism and stochasticity. Models can do this embedding random variables into their architecture, and operating their simulated processes in non-deterministic ways when executed.

Agents and networks

Most times interaction schemes are the tools for representing biological complexity. In this perspective, nodes model elements, and arcs relations between them. The underlying formalisms, generally graphs, are naturally prone to formal analyses for extracting information, and this is desirable in computational biology. Networks usually perform substantial abstractions from the mechanistic explanations of functionality, and this is a limitation.

On the other hand, agent-based models treat biological elements as separate, reactive and information-rich entities, which interact with the environment and other actors. The representation strategy, in this case, centers on states and their evolution, and the interaction scheme may be flexible. The behavioral model within agents can be either based on linguistic rules or on mathematical models such as ODEs systems. This method supports less abstract models that can represent mechanisms of interaction, besides the existence of interaction alone. This option is more computationally expensive, but it makes a more significant part of system complexity emerge in representations, covering the dynamic evolution of the system as well.

Robustness and parameter sloppiness

As pointed out in Gutenkunst et al., 2007b, a peculiarity of systems biology models is that they are incredibly robust to parameter perturbation. Authors make different hypotheses, among which one is explaining parameter sloppiness with biological robustness. That is, evolution made biological systems resilient to environmental or stochastic perturbations by lowering their sensitivity to parameters variations. According to this view, models of biology may have sloppy parameters because they are correctly recapitulating biological robustness.

Mathematical and computational

In Bartocci and Lio, 2016, authors analyze the dichotomies between mathematical and computational models.

Traditionally, formal approaches to biology base on mathematical formalisms. For example, different types of laws and systems of equations provide formal and quantitative models of biological phenomena considered in the continuum. They can cover a limited amount of system complexity before solving them becomes computationally too complex. They most times stick to single system levels or individual bioprocesses. Also, they tend to aim to universal laws holding virtually for all instances of a phenomenon. However, biological knowledge is diverse, and it organizes in a way exceptions outnumber norms, and information has both quantitative and qualitative forms. Other mathematical tools, such as graphs, are more accessible to automatic analysis and provide formal representations of interaction schemes. They perform substantial simplifications and abstractions over the system though, including in its description only functional relationships.

Computational models, in the presented perspective, better respond to the representational requirements of systems biology. On this side, building up a biological model is similar to developing a computer program. Domain-specific programming languages, with either textual or visual syntax, help modelers describe biological processes in terms of execution sequences and control flows. The semantics of these languages points to the way a computer should execute such sequences. These approaches lack the cohesive formal framework mathematical models belong to, but can more easily express the peculiarities of biological systems, which often live out of a systematic set of norms. Also, they can recapitulate the mechanistic explanation of biological functioning. Discrete, event-driven models, in particular, have smaller computational costs when executed.

A possibility to enhance model performance is to reduce complexity. On the other hand, when to scale up model complexity is desirable, computational sciences can function as enablers putting at the disposal of model executions highly parallel computing infrastructures or computational power in general. Advanced computation technologies may soften the constraints posed to models in reason of their computational complexity.

2.5 Systems biology as a research domain

The source of complexity in computational systems biology relates not only to the system. In fact, besides system complexity, models need to take into account all the complications deriving from the diversity characterizing systems biology as a research domain. Many scientific fields contribute to produce and organize knowledge flowing in systems biology. Diversity in systems biology affects both the formalisms and the modeling process, and, on another level, interactions between the professionals involved.

On the representational side, diversity implies information comes under different formalisms, due to historical and cultural stratifications specific to each field. On the first hand, this refers to the fact a variety of research fields with disjoint histories now flow into a joint scientific effort. For example, biochemistry and cellular biology had little overlaps, and so did physiology and molecular biology, until scientific advancements joined the ends of their respective domains. Until a systemic view started to take them to the same purpose, such overlaps did not pose any representational issue. On a second hand, systems biology devises the coexistence of so-called *soft* and *hard* sciences: experimental approaches are supported by technological advancements in analytical techniques, and feed computational tools and models with data so to generate knowledge. The contaminations between these traditionally separated worlds raise complex communication issues. Also, existing modeling approaches tend to target single specific subfields of biology, creating a variety of established approaches to particular classes of biological problems. For example, physicists traditionally model physiology, and consequently, models of physiological systems are expressed frequently with mathematical formalisms.

On the side of the people involved in the scientific community orbiting around systems biology, this also reflects in a great diversity of professional profiles, each one bringing in specialistic skills, specific language, and different backgrounds while lacking others. For the scopes of this work, the more relevant cultural interface is that between life scientists with an experimental background and computational scientists. Working together, these two profiles can implement all of the desirable norms in models for systems biology. Communication is key: the different educational backgrounds reflect not only in the respective expertise. It also affects the way each professional thinks and expresses knowledge. For example, the archetypical experimentalist has a bottom-up, heuristic, uncertainty-aware and serendipity-prone approach to research, while the *cliché* engineer has a top-down attitude, assuming all variables of the identified system are known and under control.

Before biological complexity, and the fact the current understanding of it is small, this intercultural issue assumes dramatic tones. A cultural transformation is necessary for making the stakeholders effectively communicate to create opportunities out of this *impasse*. Towards the passage from multi-disciplinarity to cross-disciplinarity, it is preferable that in the future such diversity generates value for the overall advancement of the understanding of biological complexity. As the first step in this direction, this work focuses on representational tools and communication means intended as enablers for this change. So it devises as a first step the creation of a *lingua franca* for knowledge representation, exchange, and inference in systems biology.

Considering that, computational approaches to systems biology acquire new facets besides their capability to accurately predict and simulate system behavior. Not all professionals in systems biology have a background in computational sciences. To serve the whole scientific community, representational tools accessible to the diverse and complex user base of systems biologists need to take shape. Such tools should preserve their formal nature and predictive power while becoming approachable and open to non-expert users. Computational models need to function as knowledge representations that is easy to exchange with no information loss. All of these aspects affect possible approaches at many levels. For example, the modeling processes need to take into account the diversity of information and data they aim to represent. That means they need to correctly read and merge different formalisms, and make sure the resulting models are consistent. In other words, they need to support hybridity. This piles up on the requirements posed by biological complexity *per se*, making the class of multi-level and hybrid models particularly relevant for the scopes of systems biology. For what concerns the computational tools and approaches in general, the challenge is to provide quantitative and formal modeling approaches with accessibility and easiness of use.

2.6 The knowledge management moments in systems biology

This section recapitulates the requirements posed to computational models by systems biology, organizing them by the different phases of the knowledge management cycle.

Knowledge inference A computational model scope can be predicting system behavior in time, or under new starting conditions or contexts. At this aim, the more system complexity a model comprises, the more accurate predictions can get. Predictions may cover not only punctual outcomes but also system evolution dynamics, making the possibility to simulate or execute them appear as a desirable feature. Models should include more of the available knowledge on complex biological phenomena also by covering more extensive parts of the system, enlarging model scopes both as in incorporating more elements from the system, and more organizational levels, and the spatial connotation underlying it. A desirable feature is the capability to cover the temporal organization of dynamic changes in hierarchic regulation levels of the system.

Wanting to infer new knowledge from available information, a model should be capable of performing accurate predictions, comprising more of the system complexity, which comes in many forms. These forms require the model supports:

- quantitative information;
- spatiality
- multiple levels;
- multiple scales;
- dynamic hierarchy;
- stochasticity;
- scalability;
- simulation or execution (temporality);
- formal analysis;
- flexibility.

Knowledge representation Systems biology is complex as a research field too, and different stakeholders can benefit from computational models. Some of them are experts in computer sciences, other pure experimentalists. In this sense, systems biology comprises a diverse population of stakeholders, which manifests over many axes. For example, users of computational tools and models may be expert computer scientists as well as pure experimentalists. Among computer scientists there may be theorists as well as pure developers, and among experimentalists the cellular biologist may work together with the physiologist and the clinical researcher. Different types of intelligence are involved in managing knowledge in systems biology: a model should be easily understandable by a human, but by a computer as well. All of these profiles have a different understanding of biological complexity, and need suitable representational tools and styles. At the same time, there is the need for a uniform representational standard. At the aim of responding to the diversity of actors involved in systems biology, while preserving a degree of uniformity, models need to allow:

- diversity-aware accessibility;
- flexible abstractions;
- usability and re-usability;
- human-readability;
- machine-readiness.

Knowledge exchange The first step for systems biology to benefit from the contributions of a diverse base of users is the construction of a *lingua franca* making different professionals able to communicate. Models can play this role functioning as knowledge bases, but only if exchanging knowledge through them is possible among all actors involved. Modeling approaches should then support:

- standardization;
- modularity;
- compactness and cohesiveness;
- interoperability;
- platform compatibility;

These requirements considered together guide the high-level objectives of this work, which the following section briefly introduces.

2.6.1 High-level objectives: a powerful and accessible modelling framework for complex biological processes

The work presented in this thesis aims to respond in a consistent and scalable way to the requirements posed by a systems biology approach to biological complexity. A comprehensive computational framework tackling the open challenges requirements draw includes:

- a modeling strategy based on a formalism which covers requirements from both system complexity and domain diversity;
- methods for automatically combining hypothesis- and data-driven approaches;
- languages, tools, and interfaces making models accessible;

The following section provides a map of the current modeling approaches and computational tools capable of satisfying one or several of the requirements posed by systems biology, so to highlight their strengths and limitations, and to furtherly define open challenges in the field.

Chapter 3

Background

This Chapter aims at mapping the scenario of existing modeling approaches to systems biology, in order to contextualize the presented solutions.

3.1 Mathematical and computational models

First of all, it is necessary to clarify the difference between mathematical and computational models and provide the respective definitions.

In Fisher and Henzinger, 2007, the author distinguishes between two types of models, mainly by the use they make of computational capabilities:

- models using ”computer power to analyze mathematical relationships between quantities”;
- models ”resembling a computer program,” and using computer power to execute the corresponding instructions.

At the aim of presenting more detailed definitions for these two classes, it is necessary to provide a more formal description of the underlying paradigms, which revolves around the difference between operational and denotational semantics.

3.1.1 Denotational semantics

A mathematical model's primary semantics is *denotational*. According to Tennent, 1976, ”in mathematical logic, a semantic interpretation for a formal language is specified by defining mappings of the syntactic constructs of the object language into their abstract meaning in an appropriate mathematical model.” For example, this could be the case of an equation-based model, drawing formal relations between biological quantities and variables, and between their change rates and parameters in a system of equations. This type of representation does not directly imply an algorithmic strategy to solve the system, and constitutes a model *per se*.

The transfer function, that is, the formal representation of the quantitative relations between system elements, is the core concept underlying mathematical models. It may be specified, for example, by an equation describing the relation between the input and the output of a system. Composing transfer functions allows describing networks of interdependent quantities, defining a class of more complex mathematical models.

For example, in Carbonell-Ballestero et al., 2014 authors cite a re-definition of the transfer function in synthetic biology as the ”response of a regulable genetic device in the presence of a signal that acts as the control variable of the system.” In this work, to characterize a set of Lux homoserine-lactone-inducible genetic devices, they experimentally determine the transfer function of underlying enzymatic reactions. Combining several building blocks of this kind, a network model for a complex metabolic system, like that described in Semenov et al., 2015, can take shape.

3.1.2 Operational semantics

On the other hand, the primary semantics of a computational model is *operational* Plotkin, 1981. That is, a model works like a computer program, whose properties undergo verification by proving the logical correctness of its execution steps and procedures.

The fundamental entity of computational models is the state machine. Its function is to link different states, intended as qualitative configurations. A simple implementation of a state machine may devise a computer program defining how a state changes into another given specific external conditions and anterior events. Complex computational models take shape from the combination of different state machines, and it is possible to define them as reactive systems Harel and Pnueli, 1985.

A reactive system is a tool to predict the emergent behavior of a biological system by the separate functioning and local interactions of state machines. Separate components represent biological entities, such as cells, as capable of undergoing state transformations after discrete events involving themselves and their neighbors. There is a shift in the perspective of a computational model over the system compared to a mathematical one: the latter considers variables evolution in terms of rates of change, while the first consider biological processes in terms of cause and effect Fisher and Henzinger, 2007.

Since, at their core, computational models consist of sequences of operations, their computer implementation naturally corresponds to the model itself. This does not hold for mathematical models, which at the contrary exist before they translate into sequences of operations. Following this difference, algorithms solving mathematical models come with a measure of their performance in terms of precision compared to the mathematical expression. Computational models need to maximize precision too, but under a different acceptation: they need to provide proper representations of system parts through abstractions.

Considering this, it is possible to build up a computational model of a biological system which is not a mathematical model, and viceversa. For example, a computer program treating cells as objects passing messages between themselves can be directly executed, and not necessarily uses mathematical formalisms to represent the system. An ODEs system represents the biological phenomenon per se, and not necessarily needs to be solved computationally. More often, computational models rely at least in part on mathematical formalisms to provide quantitative and formal representations, and mathematical models leverage computer power for simulation and solving.

3.1.3 Model analysis techniques

The differences outlined so far between mathematical and computational models imply different strategies in using them for characterizing the modeled system and its dynamics.

Solving and simulating mathematical models Algorithms can be set up to simulate or solve mathematical models. The model describes relations between quantities, and their change rates, and the algorithm simulates the corresponding dynamics. However, the relation between a model and an algorithm solving or simulating it is not exclusive. Many different algorithmic approaches may target the same model for solutions and simulations, exhibiting different performances in terms of precision.

If the constraints for individual transfer functions are relatively simple, for instance, in the case of linear differential equations, then mathematical models are amenable to analytical solutions. In other cases, analysis alone is not enough, and it is necessary to simulate models computationally to study and plot variables evolution over time. This necessity emerges for nonlinear or stochastic differential equations and models with high dimensionality.

Executing computational models Computational models, as in Fisher and Henzinger, 2007, can be defined as computer programs prescribing the step-by-step behavior of abstract machines and their interactions. As opposed to mathematical models, then, they are inherently executable. When it comes to biological applications, execution involves a large number of states, as well as non-linear and non-deterministic behavior. For these reasons, computational models for biology are usually not amenable to mathematical analysis, but they instead undergo analysis through execution. For example, some methods first developed within computer science, such as temporal logic, model checking, and runtime verification serve to this scope, and they can apply to systems biology models. These methods can help to ponder models and analyzing them in the first place. Also, they are useful for validating experimental results from the laboratory, and for checking behaviors of interest in an automatic way, as a form of pattern identification. Finally, they can automatize the input or parameter identification process for the system of interest.

All of these practices can follow the scope to determine, in the following phase, a desired behavior in the system Bartocci and Lio, 2016.

3.2 Existing computational approaches to systems biology

Computational models, thanks to the fact they are directly executable, can comprise more biological complexity than mathematical ones, minimizing computational costs. Computational time is the limiting factor when modeling large and complex systems. Respect to the intention to scale up models to vast portions of biological systems, the computational time is the limiting factor. This section focuses then mainly to computational modeling approaches to biological systems.

A variety of approaches live under the definition of computational models for systems biology. Some of them are more abstract, some more detailed. Some of them focus on the structure of a process and some others on the spatial organization of a system.

It is necessary to specify that, in systems biolgy, computational tools comprise and support models, but they make a larger category in general. In fact, computational tools include all of the systems and methods leveraging computer power for targeting biological complexity. This ranges from bioinformatic pipelines to image analysis toolkits, from model construction techniques to data visualization. In the following sections, the focus stays con computational models intended as representations of biological systems.

What characterizes computational models is they specify sets of instructions that, once executed, recapitulate a system's behavior. Different languages can describe the instruction sequence modeling the successive events in biological processes. It is possible to differentiate such languages by the way their *syntax* organizes around different representational priorities. That is, each of them has a specific set of rules for creating well-formed instructions by combining symbols.

In computational biology, most languages are domain-specific: they fit the needs and specificities of a subdomain of life science, reflecting the way that field traditionally specifies the information. Languages can also group around representation styles: some are textual, some others also have visual representations, making them more close to diagrammatic representations from biology. Visual supports help the non-expert user to access knowledge and easily manipulate information.

On the other hand, the *semantics* of a language attaches biological meaning to well-formed instructions, allowing to describe the biological behaviors of interest and provide the model execution with the capability to support biological interpretations.

The following part provides an overview of the approaches currently in place for biology, presenting each approach in reason of its general functioning and some specific applicative examples.

To different degrees, and somehow depending on their level of abstraction from calculation processes, these languages function as intermediate models.

Most times, it is possible to encode a formalism into the others. The real property guiding the choice of a formalism is its expressivity when modeling a problem under a particular perspective. Their respective syntaxes make them more or less prone to express biological semantics in different ways, being capable of modeling biological systems under different, often overlapping aspects. In other words, it is their expressivity for a particular facet of biological complexity that clusters them together in sub-classes. Beyond biological meaning, languages usually get translated in other computational models, with less abstract semantics, for being executed.

3.2.1 Agent-based models

Agent-based models Bonabeau, 2002 center on agents, that is, autonomous entities sensing the environment and making decisions according to their individual and specific sets of rules. Groups of agents can interact with each other following the same paradigm. A group of agents and their relationships define an agent-based system, which, even in its elementary forms, can exhibit complex emergent behavior patterns such as competition and collaboration.

Every agent in the system is an explicit representation of an individual. Each agent has a unique functioning and individual history, supporting, even in more complex agent-based systems, learning and adaptation of the single agent.

In modeling biological systems, agents have cellular functional/structural features and behavior. Usually, agents modeling cells express cellular behaviors and possible evolutions, as well as physical and mechanical properties.

Moreover, the agent-based system models the interactions between cells. These characteristics support a very close representation to the physical system and enable to reenact behaviors at different levels and scales, covering emergent behaviors encompassing all system levels.

Agent-based models can be implemented with tools such as FLAME Coakley et al., 2012; Richmond et al., 2010, REPAST An et al., 2009; North et al., 2006, and SPARK Solovyev et al., 2010.

3.2.2 Process Calculi

As told in Baeten, 2005, the history of Process Calculi (PC, or Process Algebras, PA) dates back to decades ago. As the names suggest, the specificity of this language is the algebraic formalism it uses. PAs are text-based languages, and their syntax uses symbols and rules from algebra and mathematics in general. The author defines PC as "the simplest model of the behavior of a computer program in computer science." PC orients to the formalization of concurrent processes, where a process is an abstract representation of partial observations over the behavior of a system. In this context, concurrent processes can also match the definition of agents. Agents can model biological species and their mutual interactions in a biological system. PC supports compositionality: starting from the specification of subprocesses, a model of the entire system can take shape, by composing them following rules dictated by the formalism. Another definite advantage of this approach is the treatment of comparisons between processes as equivalences, which permits to reason in formal terms around the relations between different subprocesses in a biological system. The primary usage of PC in biology is to formalize and organize knowledge. Some examples of PC applications in systems biology are: the Bond-Calculus Wright and Stark, 2018, Beta-Binders Degano et al., 2006; Priami and Quaglia, 2004, 2005, BlenX Dematte et al., 2008a,b, Bio-PEPA Ciocchetta and Hillston, 2009 and BioShape Buti et al., 2010.

Among the others, some implementations express spatiality and compartmentalization of the biological system. An example is Brane calculus Danos and Pradalier, 2004, which centers the simulation around biological membranes, which play the role of coordinators for the modeled processes. Another one is BioAmbients Regev et al., 2004, which is provided with special operators able to specify merging, splitting, and communication between biological compartments, and derives from pi-calculus Bruni and Montanari, 2017; BAM Muganthan et al., 2008 is a tool supporting stochastic simulations in BioAmbients.

3.2.3 Rule-based modeling

Rule-based models Angelov, 2013 are very abstract representations which focus on the rules underlying the system's behavior. They are particularly useful when the set of such rules is way more straightforward than the model it generates: the model is the enactment of a limited number of patterns repeating themselves.

This language is particularly of use for modeling certain types of biological systems. Its notation is very similar to that employed for representing chemical reactions and biochemical interactions between molecular species. They can easily cover, for example, reaction stoichiometry, and kinetic parameters of interaction.

Rule-based systems are very compact: each rule is an independent unit, making it easy to modify independently. Compactness helps accessibility: their simple syntax makes them human-readable, and possibly visually represented and modified with graphs. Existing tools for systems biology leverage these advantages, becoming accessible also for non-expert users. Some examples are BioNetGen Harris et al., 2016 , BIOCHAM Calzone et al., 2006, Kappa Wilson-Kanamori et al., 2015 and Virtual Cell Schaff et al., 2016.

3.2.4 Statecharts

Statecharts are easy-to-use, visual and state-centered formalism recapitulating the representational style of state diagrams. Passages from one state to another are event-driven, and each state corresponds to a particular set of parameters for the system. They support graphic visual representation, with the possibility to quickly highlight the interdependence between states in a reactive system.

In systems biology, statecharts are useful for they formalize the visual representations of functional diagrams widely employed in biology. Moreover, they augment them providing depth, hierarchy in state transitions and orthogonality between states Harel, 1987. These formalizations and refinements allow capturing more of the biological system complexity, limiting the risk the number of possible states explodes.

The most popular statechart tool for systems biology is IBM Rational Rhapsody Bloch and Harel, 2016; Efroni et al., 2007; Harel, Setty, et al., 2008; Hoffmann, 2012; Swerdlin et al., 2008.

3.2.5 Boolean and Qualitative Networks

In Boolean networks Dubrova et al., 2005, nodes can assume one out of two states. Each node is a boolean variable, updated by a boolean function determining its truth value given the inputs from the neighbor nodes in the network.

The most common application of this formalism to biology is the approximation of the dynamics of genetic regulatory networks. In these models, genes can be active or inactive, and boolean functions model regulatory relations between genes, supporting the simulation of their dynamic evolution Vasciaveo et al., 2015. Boolean networks can model post-transcriptional regulation as well, and this provides an higher resolution in representing biological complexity Benso, Di Carlo, Politano, Savino, and Vasciaveo, 2014; Politano, Savino, et al., 2014. However, this approach performs a strong abstraction from the complexity of the system, removing all quantitative aspects of gene activation, including the intermediate passages through gene products regulating target genes, and their respective kinetics. When dealing with large regulation networks, this can be a strategy for complexity reduction. Also, it finds applications in studying the robustness and stability of gene regulation networks. Qualitative networks extend Boolean ones, devising a finite number of states each node can assume.

Other approaches rely on graphs to model multi-omic enriched networks, including more of the different information from the system Benso, Di Carlo, Rehman, et al., 2013; Politano, Benso, Savino, et al., 2014; Politano, Logrand, et al., 2017; Politano, Orso, et al., 2016.

Graph-based coexpression networks built from experimental expression data are in general large and complex. Methods exist to reduce the complexity of these models, preserving only the most valuable part of the information they encode Benso, Cornale, et al., 2013.

Network models support multiple types of knowledge extraction procedures. The automatic identification of network motifs corresponding to identifiable biological functionalities is one example of thatDi Carlo et al., 2013; Natale et al., 2014; Politano, Benso, Di Carlo, et al., 2014. In other cases, graph-based models can support prediction of disease Benso, Di Carlo, and Politano, 2011.

Interesting examples of how Boolean networks can be used to analyze regulatory networks in systems biology are GINsim Naldi et al., 2018 and BoolNet Mussel et al., 2010. In R.-S. Wang et al., 2012 a review of the existing approaches is provided. Bio Model Analyzer Benque et al., 2012 provides an implementation of qualitative networks.

3.2.6 Petri Nets

Petri Nets (PNs) collect many advantages of the previously described approaches for modeling distributed, concurrent processes. Also, they have an exact mathematical definition of their execution semantics and support visual representation. They can easily encode process calculi and agent-based systems. Also, they can specify architectures recapitulating graph-based models such as qualitative and boolean networks. PNs can also extend network-based formalisms by including quantitative aspects from the system. They support both qualitative aspects, encoded in the network architecture, and quantitative information, in the quantification of resources and emergent network evolution.

These features are particularly useful for modeling biology. In fact, on one side PNs recapitulate and expand the expressive power of all the other formalisms. On the other side, they allow to flexibly comprise in a model the different information characterizing systems biology as a knowledge domain.

PNs come in different shades, from the low-level formalism, providing semi-quantitative discrete representations of concurrent processes, to different high-level formalisms, supporting continuous information, timings, stochasticity, and hierarchy. Section 4.2.2 provides a more thorough description of existing Petri Nets formalisms.

3.2.7 Spatial models

Modeling formalism explicitly expressing the spatial features of a biological system can cover multiple system levels. For example, they can be set up to represent microscopic, mesoscopic and macroscopic levels, accounting for molecular interactions and networks, cell-cell and cell-environment communication Drasdo et al., 2018 and tissue- or organ-level phenomena. Compared to compartment-based models such as BioAmbients Regev et al., 2004, which in a way also express spatiality, these formalisms provide a structure representing positions in space independently of the objects possibly occupying them.

Lattice-based models Checa and Prendergast, 2010; Tronnolone et al., 2017 base over a regular repeated graph, formed by identical n-dimensional grid sites. They have periodic or fixed boundary conditions in each direction over the grid.

Cellular automata Schiff, 2011 are n-dimensional grids devising, for each position, either the presence or absence of a cell. Each cell has neighbors, and according to a mathematical function taking them as inputs, the whole model evolves in terms of state changes at each position. This scheme of functioning allows to model pattern formation according to short- and long-range interactions between cells Deutsch, Dormann, et al., 2005.

In multiscale models of these kinds Alemani et al., 2012, the challenge is to set up a homogeneous representation, including communication between different model levels across multiple spatial scales. It is also necessary to reconsider specific asymptotic techniques for the analysis of the multiple time scales involved. Cellular Potts models Durand and Guesnet, 2016; Svoboda et al., 2018 combine the Monte Carlo method with a regular lattice-based model of spatiality. In this context, Osborne, 2015 provides an example of a multi-level model showing homogeneity. In general, cellular Potts models devise objects living in the lattice. These objects may be either discrete such as cells, or continuous, such as molecular gradients. Either way, their interactions, such as cell-cell communication, or cell-nutrient contact, are associated with an energy description. Energy minimization of a Hamiltonian function drives lattice rearrangements to simulate the evolution of the system, including its spatial architecture.

CompuCell3D Palm and Merks, 2015; Swat et al., 2012 is a general modeling framework for cellular Potts models, which combines rigorous energetic and mechanical consideration of the system with usability and biology-centered representational capabilities.

Lattice-free models, on the other hand, represent spatial features of a system without specifying a spatial scaffold external to the system. For example, vertex models Fletcher et al., 2013 represent cell membranes as a set of polygonal points. Basing on tensions deriving from cell-cell adhesion forces and cell elasticity, during the simulation they update the position of each vertex.

3.2.8 Hybrid Models

Hybrid modeling approaches integrate state-based, event-driven discrete formalisms presented so far with the capability to represent continuous dynamics in each modeled state. Mathematical formalisms such as ODEs usually support this feature. Hybrid systems can leverage the advantages of both mathematical and computational models, moving over the trade-off between expressivity and computational cost. They can accurately represent continuous phenomena with some model structures, and perform stronger abstractions, through discretization, in others Antsaklis and Koutsoukos, 2003; Witten et al., 1987. For this reason, they are becoming relevant to systems biology, also in terms of adaptation of dedicated experimental procedures and knowledge exchange standards Nakamura et al., 2018.

Matlab and Simulink support the design and simulation of hybrid models for systems biology Sanfelice et al., 2013. Some other examples of tools supporting hybrid modelling approaches are BioDivine Barnat et al., 2009 Breach Donze, 2010b dReach Kong et al., 2015 and Rovergene Bogomolov et al., 2015. S-TaLiRo Annpureddy et al., 2011 is a Matlab toolbox for identifying trajectories with minimal robustness in hybrid systems simulation.

3.3 Limitations of existing approaches

The different formalisms presented in this section reflect a particular approach to the modeling of biological systems. On the overall, this limits each approach to the perspective it develops around.

Each approach satisfies only partially the requirements presented in Section 2.6. Approaches limit not only to specific system levels but also to specific views over the system. In other words, each formalism centers over a particular facet of biological systems and behaviors. Its expressivity dedicates to that facet, and this impedes to express with the same effectiveness the other ones.

Compared to the others, hybrid models have a more flexible approach to biological complexity. They can fit each system part with a dedicated modeling style. Among existing hybrid approaches, the tendency is to be problem-specific and not extendable to other modeling challenges. In general, there is a lack of standardization. Hybrid models leverage the combination of different, context-specific formalisms, chosen in reason of their expressivity towards a specific aspect. Even if this has, on the overall, more expressive power than individual formalisms, it is an only methodological improvement, and it still lacks the generality, unification, homogeneity, and portability a *lingua franca* for computational systems biology needs to have.

The presentation of this Ph.D. work starts from an introduction to existing hybrid approaches which also express hierarchy (see Section 4.2.1), for then presenting a multi-level and multi-context modeling approach which relies on a single formalism, which combines several advantages of the ones presented in this section.

Chapter 4

A multi-level, multi-context modeling approach

4.1 Short summary

Starting from modeling and data management requirements, this Chapter introduces the Nets-within-nets-based modeling approach object of this Ph.D. work, and how it responds to the challenges systems biology poses. The proposed modeling approach responds to knowledge inference and representation requirements by combining the strengths of state-of-the-art solutions. Relying on the Nets-within-nets (NWNs) formalism, it provides the definition completeness of mathematical models while allowing for direct execution like computational models. Supporting different information specifications, it allows for model composition processes like hybrid models but preserving formalism uniformity. Also, it supports hierarchy and flexible abstractions. These capabilities support the construction of multi-level and multi-context models: they represent not only different organization levels from the system but also different views over them. Thanks to this versatility it is possible to model in an explicit way the role of the spatial and process contexts respectively over the system at each level. Thanks to the fact these features are made explicit, the landscape of regulations and their dynamic hierarchy emerges during execution. The proposed approach initially develops to target complex biological processes such as ontogenesis. At first, an application example for developmental biology, the Vulval Precursor Cells specification in C. elegans, is provided. It shows excellent flexibility in representation capabilities. Two more application examples follow: the first one targets a cultured synthetic biological system and the second one focuses on the spread of antibiotic resistance within the microbiota. A limitation is that models following the proposed approach work as knowledge bases only for researchers with a background in computer science.

4.2 Introduction

As in Section 2, an excellent computational model for systems biology must handle different scales of representation, and the complex hierarchical structure of the system and its sub-parts, as well as the different types of information and data available, together with their representations based on different formalisms. In other words, they need to respond to the requirements emerging both from system complexity and the heterogeneity of systems biology as a research domain.

At the moment, combining different existing methods in hybrid models (see section 4.2.1) is a strategy to obtain this result. This composition-based approach fails to respond to at least one requirement, that is, modeling approaches should be easily generalized to different systems.

The presented approach aims to unify the capabilities of existing approaches into a single one, while as well as preserving formalism uniformity, and thus enabling model flexibility, generality, and portability. For achieving this goal, we chose to rely on Nets-within-nets (NWNs), a high-level Petri Nets formalism.

In order to present the proposed modeling approach, first an overview of existing hybrid and multi-level modeling approaches (Section 4.2.1), and then an introduction to the Petri Nets formalisms and NWNs (Section 4.2.2) are provided.

4.2.1 Hybrid and multi-level modelling approaches for systems biology

Multi-level models describe a system specifying at least two different organization levels and the interactions within and between those levels Adelinde M. Uhrmacher et al., 2005. They explicitly represent upward relations, since the system is somehow constrained by the behavior of its parts, as well as downward relations, since the behavior of each part is influenced, in turn, by the behavior of the system as a whole. Considering different system levels implies considering all the different dimensional scales implied for each level, and their consistency across levels.

Multi-level models can also be hybrid models. According to Stephanou et al., "in its most general definition, a hybrid model corresponds to any interaction or coupling between two or more models that are not based on the same formalism" Stephanou and Volpert, 2016. Then, multi-level and hybrid models find a definition in the fact that they support different formalisms while being organized in multiple levels, and encompassing multiple systems scales.

In this section, the presentation of existing hybrid and multi-level modeling approaches organizes around successive moments of the modeling process.

Data and information gathering High-throughput analysis techniques make it possible to extract vast amounts of heterogeneous data from different organization levels in biological systems Davidsen et al., 2016. Directly extracting useful knowledge from them is not straightforward. Also, when data-driven and hypothesis-driven approaches work together, existing information from the system can contribute to representing and inferring new knowledge. Thus, it is necessary to store and organize the large quantity of data made available in a way that makes it ready for model integration or construction. This task needs to take into account biological data are heterogeneous. Also, a model may benefit, for example, from the combination of biological and physiological data.

Model integration and construction depend under many aspects of the context and the type of data and information in use. However, in general, they benefit from standardization and general usability of the available information, which also relates to accessible storage, unique information sources, and data reliability. These aspects can improve out of specifying dedicated practices and formalisms for data storage, and of making data and information collection systematic.

Data diversity can originate from the consideration of different system levels in a model. For example, within the Physiome Project, insilicoDB collects experimental physiological data, e.g., time series and image-based morphological models Y. Asai et al., 2011. The information contained in this database can function as the basis for the construction and simulation of biological mechanisms that occur at several system levels.

Another example of systematic data aggregation appears in Mosca et al., 2010, where the potential of a multi-level approach to breast cancer control focuses on the integration of molecular information and functional descriptions at the organ system level. With an eye to data integration, this platform also makes it possible to query existing ontologies and perform analyses and modeling of stratified data.

Other challenges emerge when dealing with data from different *-omic* pools and different system levels. Extracting cohesive information from the complete exploration of genomic, epigenomic, transcriptomic, proteomic, metabolomic, and phenomic data of a system is not trivial.

In Rajasundaram and Selbig, 2016, the authors review a series of integrative inference and analysis techniques for *-omic* datasets generated by different levels of the system. They mainly take into account interrelations and correlations between two levels, and on co-regulation analyses. In this approach, a temporal series analysis can target time-resolved experiments, focusing on how the disruption of a system spreads from one level to another. This approach can extend to populations of organisms that adapt to the different environmental conditions that affect their regulatory state.

In life sciences, sometimes data relevant to a given phenomenon is not available. For example, the kinetic parameters of metabolic reactions are limited to equilibrium states and do not cover the dynamic evolution of the system towards equilibrium. Moreover, the experimental data available in biology often refer to *in vitro* studies and not to the *in vivo* system, where the cellular microenvironment, external adaptations, and interactions with other cells influence the parameters to be studied. This lack of data is often due to technological limitations, that is unlikely to change soon. Consequently, it is necessary to develop alternative strategies for the investigation of biological complexity. Addressing limitations of data availability is not trivial; interested readers can refer to Bulik et al., 2009 for an overview of modeling approaches that address current limitations of data availability.

Model construction, integration and composition The construction of a model can start from a deductive, hypothesis-based process as well as from an inductive, data-driven process J. Cox and Mann, 2011; Fisher and Henzinger, 2007; Leonelli, 2012.

A peculiarity of multi-level and hybrid models is they often take shape from existing models by compositional processes Duncan et al., 1998. Such models are usually already complete and validated. The result of their composition, that is, the resulting comprehensive model, needs to be tested again, to ensure consistency between both the model and the system, and within the composed model.

In building up hybrid models, formalism selection and semantics specification are usually demanded to the available existing models, each of them being consistent *per se* in solving specific issues in independent and different ways. The real challenge in hybrid compositional processes is integrating such existing models.

Also, when considering multi-level models, consistency needs to exist within each level, and across different levels in the model. Existing multi-level models often deal with two (or three) levels of organization: a *micro* (a *meso*) and a *macro* level. It is possible to describe relations across levels as upward or downward causations A. M. Uhrmacher et al., 2007.

to dA number of strategies do exist for representing how sub-parts of a system at the micro-level do influence the system as a whole at the macro-level, and how the system as a whole does influence its parts Maus et al., 2011.

Formalism selection Another way to face model construction starting from existing models and heterogeneous data is to build up a brand new model, re-specifying and combine existing ones using a single formalism.

For example, DEVS (Discrete Event Systems Specification) is a formalism supporting this modeling strategy. In its original formulation, it covers the macro level with coupled modules. These modules act as executives for the atomic models from the micro level, which represent the parts of the system Vangheluwe, 2000. This approach shows great flexibility, but it is not capable of setting global variables to the behavior of sub-models. Besides, all interactions at the micro level are asynchronous, which can create some inconsistencies both within and across levels.

In A. M. Uhrmacher et al., 2007, the authors present possible solutions to these limitations. They introduce a multi-level-DEVS formalism. First of all, at the macro level, coupled models have states and behaviors of their own. Then, consistency across levels is made stronger by a system of upward and downward exchanges between levels, which are explicitly defined and support selective communication. Cross-level communication allows to refine communication across levels: threshold crossings at the micro level can determine discrete state changes at the macro level. The macro level, on the other hand, can activate modules at the micro level specifically sending events to them. Moreover, such communication channels function synchronously.

Moving to the field of high-level model description languages, in Yoshiyuki Asai, Abe, et al., 2014 a platform for integrating two different mark-up languages is presented.

This strategy starts from the fact that both languages are usable and interoperable, enabling the combination of the respective expressive powers. The Systems Biology Markup Language (SMBL) and the Physiological Hierarchy Markup Language (PHML), in fact, present specific advantages and specificities. SBML Hucka et al., 2003 tends to model sub-cellular phenomena and translates to ODE systems to be simulated. PHML, instead, better orients to the representation of hierarchically organized systems, as its ancestor insilicoML did Yoshiyuki Asai, Y. Suzuki, et al., 2008. In this work, a PHML framework embeds modules specified in SBML. This combination extends the expressivity of SBML to several levels from the system, thanks to the hierarchical organization of PHML. A model structured in this way ensures great expressivity, but it can become computationally expensive when simulating the underlying mathematical formalisms.

Modelling approaches A typical application of multi-level and hybrid model construction processes is the study of tumor growth. Modeling tumor growth under different perspectives is possible. For example, some approaches focus on the macro level, where the tumor appears as an entity inside an organism, showing independent behaviors and relations with other entities. Another one centers at the micro level, where the tumor appears as a complex structure whose behavior emerges from local interactions between single cells, with their inner functionings, intertwined with communications between each other and the environment. It is also possible to consider a meso level, where the significant entities are cellular aggregates and their architecture mediating cell-cell and cell-environment communications LACHOWICZ, 2005.

In Delsanto et al., 2008, the authors describe a multi-level model of in vitro tumor spheroids and the effects of environmental stimuli on their growth. Cellular aggregates make the lower level in the model, while the macroscopic regulations make the higher one. The most significant contribution of this work is the construction of an intermediate model interfacing the two models. Such structure can put in the correct relations input and output functions between the levels, making them communicating in a way that is consistent with the experimental data to be modeled. Since the two bridged models stem from independent model construction processes, the fact that they can generate consistent behaviors provides a sort of mutual validation for both of them, highlighting one of the advantages provided by hybrid modeling strategies: inherent validation by direct comparison of independently developed models to be combined Delsanto et al., 2008.

Most often, multi-level modeling approaches to systems biology deal with multi-cellular systems and biological tissues. More rarely, they face the aim to model a whole organ. For example, in the context of the Physiome Project Holzhutter et al., 2012, researchers are trying to build a virtual liver. This virtual organ encompasses multiple system levels, comprising a wide range of time and space scales. For example, a hormone requires seconds to exert its action on cellular receptors. Tissue regeneration processes, take weeks instead. Single cells live on the scale of micrometers, while the whole organism the modeled organ belongs to on the scale of meters. This perspective is relevant since one of the modules composing the model refers to the whole body, modeling it with a Physiologically Based Pharmacokinetic (PBPK) model Zhao et al., 2011. The scope of this module is to consider the contribution of all body districts to the functional context for the liver to live. The virtual liver supports many different functional levels. For instance, the Perfusion module models incoming blood flow at the organ level. The model starts with the assumption of anatomical micro-homogeneity. That is, the smallest functional units are assumed to share homogeneous structures. These functional units find representation in the Lobule modules. Sets of cells compose them and recapitulate the organ function. Lobule modules function at a steady state most times. Cell replacement occurs at a decidedly slower rate than that of other processes in the model. Sometimes this rate increases. For example, when tissue regeneration occurs, cells vary in number and identity at a faster rate. In this case, single cell dynamics can determine process evolution. An agent-based approach underlies the model of cells, which then react to their environment according to their internal rules. All modules in the model are coupled, and therefore interdependent. In principle, any variable change could affect all other variables on some level. In this way, the model is a very accurate and integrated one, but it also makes a vast computational complexity emerge when simulating it.

Authors in Delsanto et al., 2008 provide another example of a hybrid and multi-level modeling approach, relying on a different strategy. Authors model tumor spheroids cultured *in vitro*, and the way external signals can alter their growth.

In the model, the bottom level represents cellular aggregates, and the top level covers the external signals. The peculiarity of this approach lies in what lies between different model levels. The authors design an intermediate model, functioning as an interface between the other two. This interface puts in consistent relations inputs and outputs from the two sides, making them communicate properly. The fact that the overall model is consistent with the biological system modeled works as a sort of validation for the two models it integrates. This example hints a collateral advantage hybrid modeling has: it provides a direct comparison between integrated, pre-existing models, working as a form of built-in validation Delsanto et al., 2008.

Parametrization and parameter identification Parametrization defines, broadly speaking, the fact that a model represents some physical quantities as static parameters, instead of computing them dynamically Godfrey and DiStefano III, 1987. In the case of mathematical models, this requires to find a set of parametric equations to describe the system.

Parameters are measurable factors defining specific aspects of the system. They are usually numerical, but they can also find other representation.

For most biological systems, the majority of the parameters are either unknown or largely uncertain. The reason a model represents a system part through a parameter, in this context, is that the phenomena it corresponds to are too small or too complex to be measured. In a different case, the model treats them as system variables. In this cases, the resulting parameters are said to be *loosely constrained*, or *ill conditioned* Moles et al., 2003.

Parameter identification is the task of estimating parameter values for a given model, usually by fitting the model to experimental data. The parametrization is generally non-unique: different sets of parameters can be used to represent the same data.

In the compositional processes yielding to multi-level and hybrid models, parametrization concerns the models to be combined together more than the resulting overall multi-level model. In fact, the compositional process does not impose a different structure to the submodels, but only draw links between them.

The number of effective parameters of a model is a good measure of its complexity Spiegelhalter et al., 2002, and this holds for composite models too. Multi-level models pose an additional challenge in this context since quite often parameters of a given model level represent variables for the corresponding upward layer Gil-Quijano et al., 2012. This layer of hyperparameters makes the identification of effective parameters, and therefore the corresponding measure of model complexity more complicated.

The more complex a model is, the wider is the set of possible trajectories in its evolution, reflecting directly to parameter identification, which can become a computationally intensive task. Breach Donze, 2010a is a Matlab/C++ toolbox for verification and property-driven parameter synthesis based on Signal Temporal Logic Raman et al., 2014. This tool is specific to non-linear hybrid models, and it is based on a very efficient numerical solver of ODEs, making it able to handle the complexity of the task.

ABC-SysBio Liepe et al., 2014 is a platform providing tools for model selection and parameter estimation in systems biology. This tool works with models written in SBML, both deterministic and stochastic, and it relies on by Bayesian computation Sunnaaker et al., 2013, which is pretty useful for inferring the parameter values of complex models, in particular when they rely on ODEs as a formalism. GNU MCSim Bois, 2009 can perform numerical simulations of different kinds, including simple runs, as well as plain or Markov Chain Monte Carlo simulations Gilks et al., 1995. Also, it supports Bayesian statistical inference for equation systems.

Besides structural relations between the parts of a biological system, parameters also model quantities relative to process dynamics, for example, the time-delays in an evolving regulatory network. What generates these delays is usually not well-characterized, and most probably a multi-factorial mechanism, making the identification of the corresponding parameters an ill-defined problem. To face this challenge, in von Stosch et al., 2010, authors present a semi-parametric, hybrid approach for performing system identification for biochemical networks with time-delays, performing better than the approaches avoiding tho consider them.

Verification and validation Verification ensures model correctness, by finding and fixing model errors, assuring that the model matches the starting assumptions and specifications Carson, 2002. For hybrid and multi-level models verification concerns also the way different models communicate between different levels and from different formalisms Tokishi and Chiu, 2013.

UPPAAL Behrmann et al., 2001 is a very usable, integrated tool environment taking care of model construction, validation and verification of dynamical hierarchical hybrid systems. It consists of different parts: two languages, one description-oriented, another one supporting multiple data types, and non-deterministic guarded commands; the simulator, which allows for validation exploring possible evolutions during the early phases of the design process already; a model-checker, which performs model verification by exploring in an exhaustive way the entire state-space of the system.

Validation makes sure that the model represents the system to be modeled at an accepted level of accuracy Carson, 2002. Techniques such as cross-validation have the objective to assess to what degree the model under validation can generalize to a data set that did not contribute to model construction.

When dealing with hierarchical Bayesian models of phylogenesis, a possible validation approach is to analyze marginal likelihoods Xie et al., 2010, but this shows high sensitivity to model priors Duchene et al., 2016. To improve the validation of these models, the authors introduce another approach, based on the expansion of the cross-validation method proposed in Lartillot et al., 2009, to include other components of the Bayesian hierarchical model in the rotation estimation process.

4.2.2 Petri Nets formalisms for biology

In the context of multi-level, hybrid models, the Petri Nets (PNs) formalism leverages different strengths from the existing approaches introduced in Section 3 Bonzanni, Feenstra, et al., 2014; F. Liu and Heiner, 2010. As an unambiguous formalism, they can easily encode most other formal notations. Petri Nets support both visual representation and complete mathematical description and analysis. The structure of PNs solidly bases on causality, allowing to discriminate between concurrent and mutually exclusive behaviors finely F. Liu, Heiner, and D. Gilbert, 2017.

In the class of PNs, several formalisms exist, and each of them can support the modeling of biological systems to different extents. Several general purpose simulation tools that allow real-time inspection and network simulation using Petri Nets are available.

In the following sections, an overview of different PNs formalisms intends to provide a context for the introduction of Nets-within-nets, the formalism the proposed modeling strategy relies on for modeling complex biological processes.

Low-level Petri Nets

Petri nets (PNs) is a graphical, mathematical modeling tool, named after Carl Adam Petri, who created the formalism in 1962 to study communication with automata.

The low-level PNs formalism supports a model covering these features, combining usability and simplicity in model design with the capability of supporting dynamic simulations and formal, quantitative analysis. For these reasons, as reviewed in Koch, 2015, low-level Petri Nets are a valuable state-of-the-art tool for computational biology.

Petri Nets at their core are bipartite, directed graphs consisting of two kinds of nodes: *places*, represented by circles, and *transitions*, represented by boxes. A set of directed *arcs* connects the nodes, usually labeled with weights that represent the minimum tokens required to trigger the transition having the place as an input. A place that has an outgoing arc towards a transition is an input place for a transition. A place that has an incoming arc from a transition is an output place. As shown in Figure 4.1, each place can contain a number of *tokens*. Tokens provide a quantitative and discrete representation of resources, and they are another element of the PNs formalism.

At each moment during net evolution, the *marking* recapitulates the position of each token in the net: the state or marking of a net is its assignment of tokens to places. The initial marking models the starting conditions for system evolution, that is, the state of the net at the beginning of a simulation. After each simulation step, the marking evolves after transitions move, produce or consume tokens in the net. The evolution of the marking along a simulation represents the system dynamics emerging along time.

Transitions function according to specific, local rules, regulating both enabling and firing. Rules define the conditions required for the transition to fire (e.g., a particular marking of the input places), and the effect of the transition (i.e., how tokens move when the transition fires). Firing transforms the current marking, and it can involve tokens in different ways: consuming them, putting them back, moving them or generating new ones to the output places, or a combination of these, according to the firing rules of the specific transition.

Net architecture organizes these rule-based functionings over the connections between input places, transitions and output places. The output place for a transition can work as an input place for another transition. In this way, several interlocked mechanisms find representation in a PNs model. Also, a transition can have multiple output places, linked to parallel downhill mechanisms, supporting the modeling and simulation of distributed systems and concurrent processes competing for resources Koch, 2015, here modeled by tokens. A PN model comprises an arbitrarily large number of these structures, each with specific architecture, rules, and connections to the other ones.

PNs easily model isolated biological mechanisms, such as biochemical reactions, representing semi-quantitative, stoichiometric relations between molecular species involved. In regulation networks such as genetic or metabolic ones, several reactions combine, and further requirements emerge. A low-level PNs model faces these requirements too. In modeling regulation or metabolic networks, places can model molecular species and enzymes from biochemical reactions. Tokens can model biomolecules in a discretized way. Transitions can model the reaction processes, covering with their rules the stoichiometry and the biochemical transformation of resources in a semi-quantitative way.

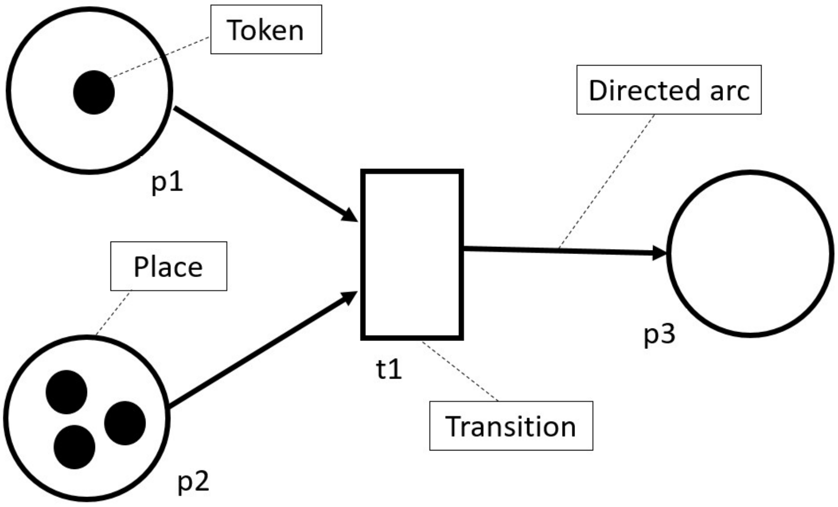


Figure 4.1. A simple Petri Net: places (circles) and transitions (boxes) are connected by directed arcs. Places represent states, and transitions processes taking from one state to another. Each transition has enabling and firing rules determining if it can fire, and when it fires given it can. Each place is characterized by the presence or absence of tokens (black dots), which represent discretized quantities of resources.

The network architecture in Figure 4.1 could model a variety of biological mechanisms, according to the semantics the modeler chooses for it.

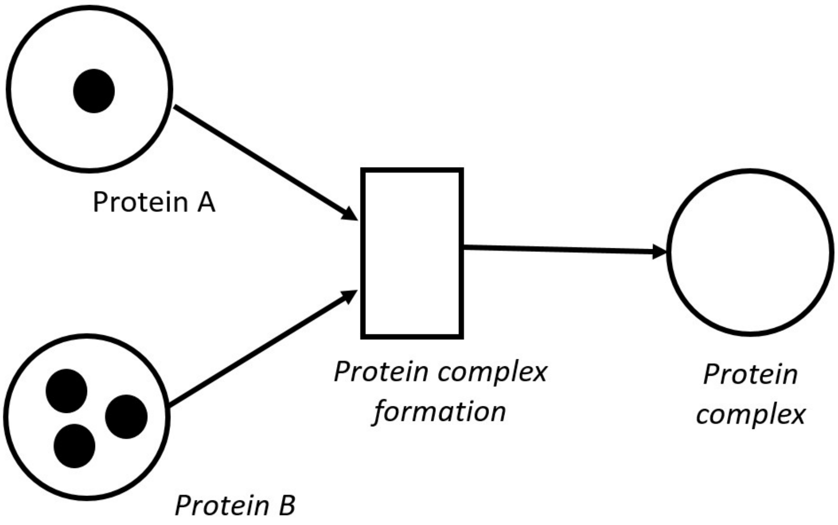


Figure 4.2. A place models Protein A, and another one Protein B. Each token represents a single instance of a protein. The transition represents the process by which a protein complex forms out of protein A and protein B. The transition gets enabled when at least one token is present in both input places.

For example, in Figure 4.2 a place models Protein A, and another one protein B. Each token represents a single instance of a protein. The transition represents the process by which a protein complex forms out of protein A and protein B. The transition gets enabled when at least one token is present in both input places. When it fires, it consumes one token from Protein A and one from Protein B respectively, and it produces one token into the place modeling the protein complex. In this way, a simple Petri Net recapitulates the stoichiometry of this biological process.

In general, low-level PNs do not scale with system complexity, restraining their use to the modeling of only small systems. Their limitations emerge when models aim to comprise more system complexity. The visual language of PNs, in fact, does not support the construction of large and complex models. It is more probable that the modeler makes a mistake, and the visual support fails to guarantee any clarity of understanding.

Complex systems need quantitative representation. Complexity not only makes the net architecture larger and more intricate. If the diversity of mechanisms from a biological system enter a model, this imposes the formalism supports different types of information, quantitative and qualitative. Also, systems biology imposes that a model provides proper representation for the multiple organization levels of biological systems and their hierarchy. In general, biological processes are intrinsically stochastic: PNs models need to express stochasticity as well when executed.

High-level Petri Nets

To overcome these limitations, several high-level PNs extend the low-level formalism, supporting multi-level and nested models that properly handle information diversity, including more system complexity into models F. Liu, Heiner, and D. Gilbert, 2017.

Colored Petri Nets (CPNs) Jensen, 1987 support the representation of arbitrarily complex data structures attached to tokens. A model supports different information structures, which take the name of *colors*, and each place in the net supports a subset of colors, limiting the token types it accepts. In each place, a multi-set over the color set attached to the place defines the marking F. Liu and Heiner, 2010. In CPNs, each token can carry structured information, allowing it to represent different types of resources. Introducing colors makes CPNs valuable visual modeling tools form complex systems as well, for they allow for non-redundant, more compact representations. Compactness improves readability and averts modeling errors while preserving the modeling capabilities of low-level PNs, which can be generated from CPNs models by automatic unfolding F. Liu and Heiner, 2010; F. Liu, Heiner, and D. Gilbert, 2017.

The Timed Petri Nets (TPNs) Scheidel et al., 2015 formalism extends the low-level PNs capability by setting specific timings for transition firing. That is, once a transition is enabled, deterministic time delays can occur before actual firing, ordering different transition activations during net evolution. Delays are tunable parameters in the model. Time delays increase the time resolution of representations, and they allow to include in models mechanisms centered at different timescales.

For modeling the inherent stochasticity in biological systems, stochastic PNs (SPNs) F. Liu, Heiner, and Yang, 2016 extend TPNs introducing probabilistic time delays. That is, time delays between enabling and actual firing are no more tunable parameters, but instead random variables. Their value can also depend on the current marking of the net, adding a representational layer for interdependencies within and across model levels. These capabilities prove useful in modeling biological systems Schulz-Trieglaff, 2005.

Hierarchical (or nested) Petri Nets can model multi-level biological systems. Representing parts and subparts in nested net architectures, they make the hierarchical relations between them explicit, allowing for arbitrarily high resolution in the description of mechanisms from different system levels Marwan et al., 2011.

Nested PNs aim at representing multi-level systems with single-level models. Also, similarly to CPNs, they stick to a static paradigm: token colors correspond to static data structures, and nets have a static model architecture. Resources can change state only by moving from place to place, and the models devise mobility for tokens but not for other model parts.

Complex biological processes challenge the limitations of most high-level PNs. They consider biological systems as dynamic structures with multiple regulation set-ups and structural conformations across different phases of the same process. Developing conformations involve evolutions of system architectural and functional patterns, including the movement and generation of new system parts, and decision making processes based on the outcome of previous process stages. This new level of biological complexity reflects into further requirements to computational models and the underlying formalisms.

The Nets-Within-Nets formalism

As introduced in Valk, 2003, Nets-within-nets (NWNs) can express all of the functionalities from other high-level PNs formalisms, such as stochasticity, timings, hierarchy, and quantitative information. Besides, they innovate PNs-based modeling strategies providing tokens with a PNs structure in turn. That is, NWNs go beyond the concept of static token color, by attaching dynamic information to tokens using the PNs formalism itself Kummer et al., 2004. Tokens specified in this way are called *net tokens*, or *object nets*. As Petri Nets, they evolve dynamically like the net holding them, which takes the name of *system net*. Also, they can hold net tokens in turn. This recursive scheme can be reiterated in a boundless fashion, allowing for open recursion in specifying the hierarchical organization of system levels with dedicated model layers Cabac, Duvigneau, et al., 2005 (see Figure 4.3).

In other words, NWNs follow a paradigm similar to that of Object-Oriented Programming (OOP): tokens in an NWNs model can take the form of classes' instances. These classes can be specified with the NWNs formalism, in turn, living within and being simulated concurrently with a higher-level NWNs model. As instances of NWNs models, object nets can hold net tokens as well, and this can be repeated recursively, specifying as many model levels as desired. This flexible structure provides full expressivity, in an NWNs model, for representing system hierarchy.

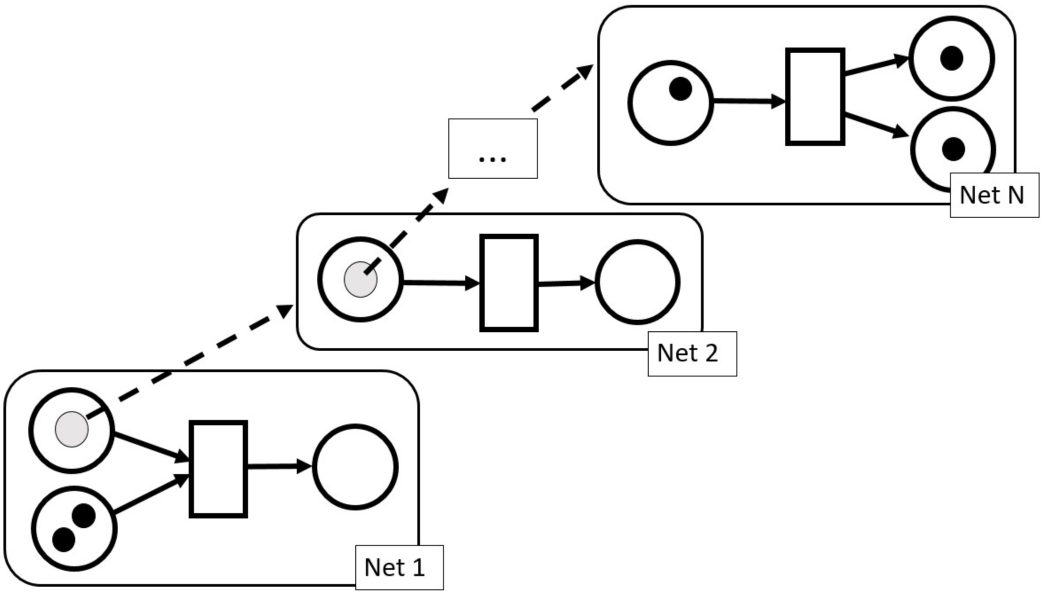


Figure 4.3. In Nets-within-nets (NWNs), tokens can be specified with the NWNs formalism in turn, generating an arbitrary number of nets within other nets. This allows to model the hierarchical organization of biological systems.

According to their definition, NWNs are particularly suited to model distributed systems that require the expression of hierarchy and encapsulation. Thanks to their capability to express encapsulation and selective communication, they can easily represent biological compartmentalization and semi-permeability of biological membranes. Moreover, by construction, the NWNs formalism recapitulates the OOP formalisms. This similarity facilitates the integration of NWNs models with several modern programming languages.

The following paragraphs introduce the features and capabilities of the NWNs formalism, as a premise for the presentation of the NWN-based modeling strategy object of this work, which follows.

Object systems In Valk, 2003, the author defines a *elementary object system* as a single system net and its marking, which comprises either net tokens, with their markings, or simple black tokens. These elements draw a hierarchy of two levels: the system net on top, and the net tokens on the bottom. In this hierarchy, net tokens are objects, instances of net classes. They can be instantiated within other net instances, creating a system of nets.

A possible extension of the concept of elementary object system considers that the same net token instance can live in different system nets. This peculiarity allows for specifying different facets of the context to be modeled for the net token.

Transports and interactions Different system nets can host the same net token, and each net token can navigate system nets following different mechanisms. Transitions in the system net can *transport* net tokens from a place to another one without determining any other changes. In this case, net tokens function independently and concurrently to the system net. In other cases, transitions from the different nets interlock: a *interaction* between them takes place. Interactions between different nets rely on communication mechanisms such as *synchronous communication channels*, which join transitions across nets Christensen and Hansen, 1994. Each channel has two ends: the *down-link* and the *up-link*. The transition containing the *down-link*, when enabled, checks for the presence of the corresponding *up-link* in the nets system. If it finds it, they activate synchronously. Transitions containing a *up-link*, on the other hand, wait for the corresponding *down-link* to evoke their joint activation. A single *down-link* can activate multiple *up-links* at the same time. *Up-links* and *down-links* define, in a sense, a directionality for channels. Channels can also pass arguments, supporting tokens flow across nets. The direction of tokens flow between the *down-link* and the *up-link* transitions is independent from channel directionality, and it can occur in both directions.

Intra- and cross-layer interactions In a hierarchical system of nets, interactions involve transitions from both the same layer and different layers. They can result in either *writing* or *reading* mechanisms. That is, the description of transitions activation can focus either on their determination of the marking in the output places or on their consideration of the marking in the input places for the next evolutions. Combining these options, four categories of communication mechanisms take shape:

- **Intra-layer reading**: a transition considers the marking at input places of another transition from the same model layer.
- **Intra-layer writing**: a transition affects the marking at output places of another transition from the same model layer.
- **Cross-layer reading**: a transition considers the marking of an input place of another transition from a different model layer.
- **Cross-layer writing**: a transition affects the marking at output places of another transition from a different model layer.

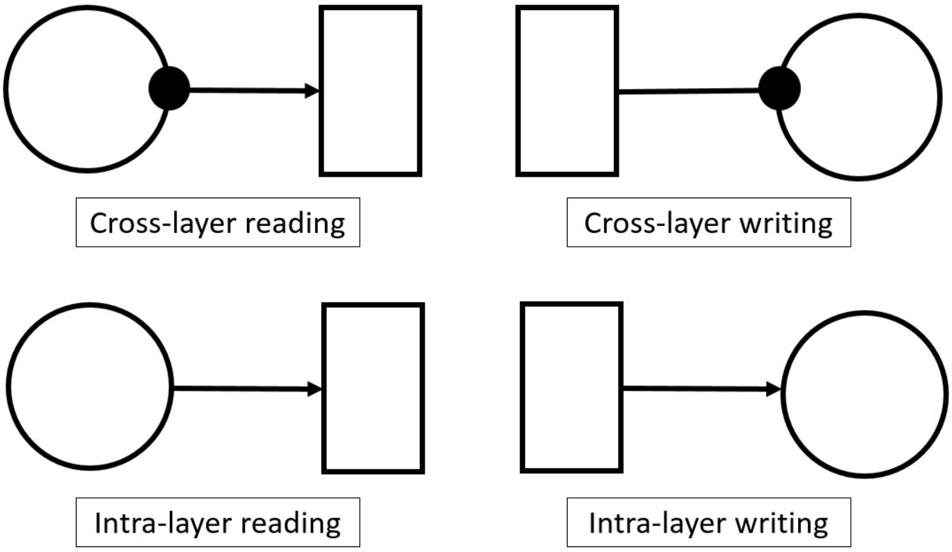


Figure 4.4. NWNs communication styles: a visual representation of the four communication styles in NWNs. Plain arrows represent *intra*-layer mechanisms, while full dots *cross*-layer mechanisms. Transitions *read* from their input places, and *write* in their output places.

These mechanisms recapitulated in Figure 4.4, allow for communication between different nets, and combining them it is possible to build up a consistent hierarchical model of a multi-level system and the contexts for its dynamic evolution.

4.3 A multi-level, multi-context modeling approach to complex biological processes

In a systems biology perspective, a good computational model should support with an equal level of detail the available information from all system levels of interest. Including this information implies to represent all the interdependent layers of regulation involved, and their context-dependent, dynamic hierarchy. In particular, both the level of cells and their dynamic regulative landscape, including their functional and process context: the model needs to explicitly represent spatiality in the system and temporal organization of the process. Also, systems biology poses general requirements to computational models, involving the capability to represent quantitative information, nonlinear dynamics and stochastic behaviors.

Computational models can have different scopes, for example, prediction and analysis. They can also serve as a very structured, information-rich knowledge base. A range of scientific subdomains with their own culture flow into systems biology as a research domain, and scientific collaborations involve a diversity of actors, each one speaking a particular language. Models need to reach this complex and diverse base of potential users, functioning as accessible tools to foster model-based scientific collaboration across disciplines. Also, they need to be scalable and generalizable, so to comprise the enormous complexity of biological systems and show high performances in different applications.

The presented modeling strategy extends the concept of *elementary object system* (see Section 4.2.2, leveraging the capabilities of NWNs-based computational models for targeting complex biological processes. The capability to express encapsulation, selective communication, and hierarchy, together with the support of colored tokens, consistently allows handling different information structures in the same model, making sure they are coherent with the semantics of each NWNs instance in the model.

This modeling strategy was primarily developed to model ontogenetic processes. This class of phenomena was the starting point for developing the strategy because it is intended as the bioprocess involving the more system levels, strongly relying on spatiality, and showing the most dynamic hierarchical relations between different regulation layers. Also, available information from these processes come in many formats, from time series describing state changes of cells to transcriptional profiles, imposing that models support different information. In other words, ontogenetic processes stress every requirement posed by biological complexity and systems biology. For this reason, they are the right target for developing a modeling strategy: by aiming at ontogenetic complexity, the resulting approach easily and flexibly applies to many other systems posing more relaxed requirements.

For this reason, the syntax underlying the proposed modeling approach develops around a semantics relevant to ontogenesis. Nonetheless, the same approach can flexibly extend to other target processes. This Chapter provides various application examples and uses case scenarios, so to show the flexibility of the presented approach. Each application example develops around some common points:

- an introduction to the **class of biological processes** to model, and the requirements it poses;
- a declaration of the **modeling objective**;
- a presentation of the **modeling strategy**: basic building blocks and model organization;
- the brief presentation of a **application example** relevant to the domain;

In particular, after introducing the original application to ontogenetic processes, this scheme is applied to the modeling of *in vitro* synthetic biological systems, and that of epidemiological processes.

4.4 Modeling ontogenetic processes

Ontogenesis (or morphogenesis) is one of the key concepts at the base of the developmental biology Scott et al., 2001, defined as the origination and ”[...] *development of a single individual, or a system within the individual, from the fertilized egg to maturation and death*” Said, 2018. Ontogenesis concerns developing embryos of multi-cellular organisms as well as unicellular life forms not having an embryonic stage in their life cycle. In this work, we focus on the ontogenesis of a multi-cellular organism, which presents emergent *architectural* and *phenotypic* complexities and takes place following *process stages*.

Systems biology targets complexity with a holistic approach, considering a system as more than the sum of its parts Palsson, 2015. Under this perspective, ontogenesis comprises a complex and intertwined processes at multiple system levels, from the development of the organism as a whole at the macroscale to the differentiation of single cells at the microscale. In morphogenesis, *emergent patterns*, at the mesoscale, are aggregates of cells with different phenotypic identities grouped following a defined spatial organization Nelson et al., 2005. Patterns reshape after each developmental stage, and changes emerge from local interactions between cells, occurring over different distances and time ranges.

Following and extending the classification proposed in Jernvall, Newman, et al., 2003, the repertoire of *basic ontogenetic mechanisms* making such complex dynamics emerge in three classes:

- *autonomous* mechanisms, making the internal dynamics of the cell and resulting outward behaviors, such as division of a heterogeneous egg, and different mitotic spatiotemporal patterns. Alternatively, the evolution of cell identity, considered in reason of its functional markers;
- *inductive* mechanisms: cells affect each others' autonomous mechanisms *via* either unilateral (hierarchical) or bilateral (emergent) signaling during pattern formation.
- *morphogenetic* mechanisms: phenomena changing the spatial architecture of cells (the *form* of tissue) in a developing structure without directly affecting their internal dynamics. Some examples include directed mitosis, differential growth, and adhesion, apoptotic and migration processes, contraction, and matrix modification.

In a developmental process, each stage corresponds to a different architecture, regulative set-up or subprocess in the organism and its subparts. Architectural conformations, as a form of *morphogenetic* mechanism, dictate the communication schemes the cells participate in Guglielmi and Renzis, 2017, setting up a scheme of relative positions between cells. This mediates cell-cell communication, that is, *inductive* and subsequently *cell autonomous* mechanisms. *Basic ontogenetic mechanisms*, as in Sharpe, 2017, are defined as ”tractable and understandable phenomena," the result of a reductionist approach to complexity, which deconstructs the system to facilitate our understanding. In this work, they are intended as *building blocks* for facilitating the construction of a model.

Rather than a linear combination of subprocesses, a multi-dimensional, *dynamic landscape* of interdependent, diverse and complex regulation mechanisms underlies ontogenesis Huang, 2012; Nepal et al., 2013. At each developmental stage, the cellular microenvironment affects *cell autonomous* mechanisms. The microenvironment defines the *context* cells live into under two main aspects. On the one hand, the *functional context* includes neighboring cells, their architecture and environmental signals. On the other hand, the *process context* refers to the stage the cell lives into at a particular moment. In some circumstances, a regulation mechanism may overtake others, but the situation can reverse when the context evolves.

In order to holistically comprise the resulting dynamic hierarchy of regulation layers, models of development need to integrate multiple system levels consistently. At the same time, reducing biological complexity to understandable phenomena allows for straightforward knowledge interpretation and exchange.

In this work, we present a modeling strategy for ontogenesis relying on the combination of both the holistic and reductionist approaches. The fundamental bricks of models following this strategy are *functional modules*, modeling *complex ontogenetic mechanisms* thanks to the following features:

- they encompass all system levels of interest;
- they can function as scaffolds for a set of *basic building blocks*, mediating the combination of their functionalities within the multi-level hierarchy of the model;
- they have abstract architecture and adjustable parameters, making them both generalizable across different ontogenetic processes and fine-tunable to specific modeling applications;
- they can be combined forming models that naturally exhibit consistency between time and space scales at all system levels.

In this way, our approach supports generalization and knowledge exchange, as well as the gain of a deep, systemic insight over the system.

In the presented approach, *functional modules* and the overall model share an essential backbone centered on two system levels:

- the cells, and their internal regulation circuitry, including all relevant *omics*.
- the dynamic regulative landscape cells live into, as in their functional and process context.

4.4.1 Model organization for ontogenesis

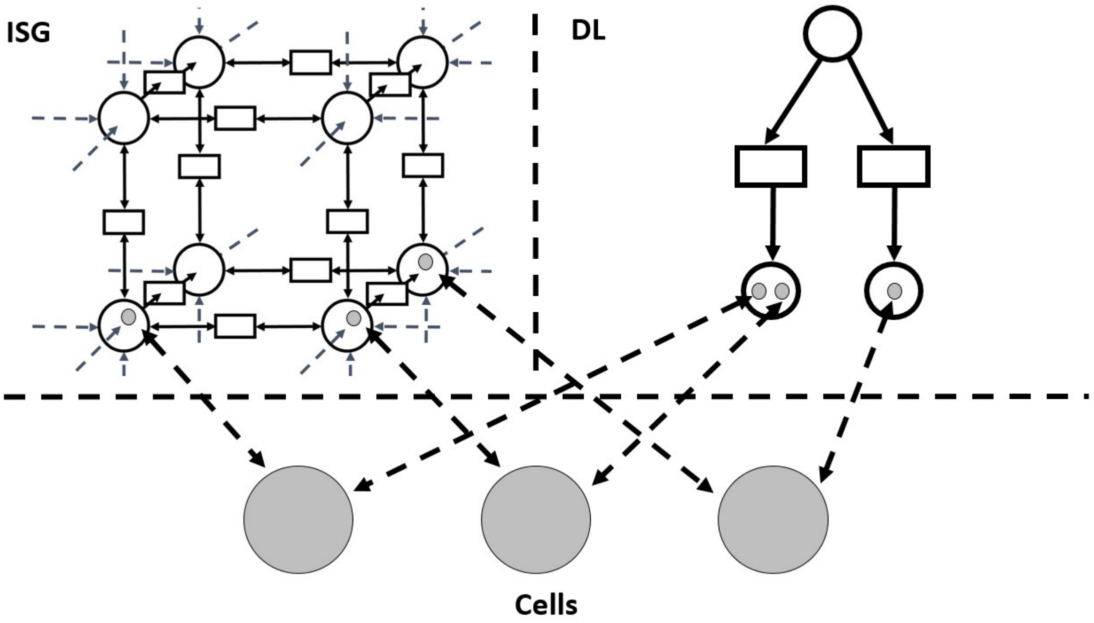


Figure 4.5. General model architecture for the modeling approach to be presented in this Section. Net tokens at the bottom level model cells with their internal dynamics (not shown in this Figure); net token instances live in the top level in two system nets: the Interactive Spatial Grid (ISG), modeling the *functional context* for the cells, and the Differentiative Landscape (DL), modeling the *process context*.

As depicted in Figure 4.5, the presented modeling approach is structured with two main levels. The top level hosts a set of two *system nets*, each one dealing with a different view over the complex regulatory landscape of the system. Net tokens populate the bottom level, each one representing one of the cells from the system. Each net token instance lives in both system nets.

At the bottom level, net tokens model cells composing the developing system; at the top level, the system nets represent their functional and process contexts, reflecting in different semantics for the different nets. Synchronous communication mechanisms make the whole model consistent.

Interactive Spatial Grid At the top level, this system net model the spatial architecture of the system, together with all possible interactions taking place between each pair of positions in the grid. It covers the *spatial context* and implications of *morphogenetic* and *inductive* mechanisms.

Differentiative Landscape This other system net from the top level models the *process context*, providing a functional scenery for *inductive* and *cell autonomous* mechanisms to take place.

Cells At the bottom level, net tokens model the cells from the developing system. They comprise *cell autonomous* mechanisms, such as cell differentiation, and the regulatory networks underlying them within the cell. They also partly model *inductive* mechanisms, for their origination or effects within the cell.

Building blocks for basic ontogenetic mechanisms

This Section describes the basic building blocks that compose the nets at each level. These structures reflect the definition of *meta-transitions* presented in Bonzanni, Krepska, et al., 2009: "[...] a Petri net with specified inputs (places which can receive tokens) and outputs (arcs outgoing of the module)”. Building blocks possibly rely on either intra- or cross-layer communication mechanisms. In the second case, the cross-level communications arrange in a way that ensures consistency between the semantics of the two levels involved.

Each building block is a general model structure, corresponding to a general mechanism underlying ontogenesis. This modular approach makes the presented modeling strategy both specific for ontogenesis and general across different ontogenetic processes. When employed for modeling specific processes, building blocks quickly adapt by adjusting their parameters.

Interactive Spatial Grid

In the ISG, the following relations between model and system elements hold:

- *Places* model subparts of space (in either one, two or three dimensions). Each subspace holds a cell plus its pertinences;
- *Transitions* model transports or interactions between actors living either in the same or in adjacent places, also marking relations of mutual neighborhood between subspaces.
- *Coloured tokens* model biomolecules in the extracellular space;
- *Net tokens* model cells;
- *Net architecture* models the grid (either uni- bi- or three-dimensional) of subspaces, and their respective adjacencies and interactions.

In an ISG, model elements form the following building blocks, modeling *morphogenetic* and *inductive* ontogenetic mechanisms. Each place in the grid has all of the capabilities described, and this makes the grid *interactive* and reactive to the net token instances it contains.

The following Figures depict building blocks presented in the main text. For visual simplicity, they highlight the most relevant net structures to explain their respective peculiarity. In particular, in illustrating building blocks from the ISG, a transition connecting two places in the grid by double-edged arcs summarizes the variety of interactions that can exist between them.

When a zoom into net instances (Cell A or Cell B) living in places from the ISG is useful for explaining how a building block works, a grey triangule works as the background for the relevant net structures of the instances.

Neighbor detection is the basic process implemented by the ISG. It marks the neighborhood relations between two positions, allowing a net token occupying a place to retrieve the identity of (Figure 4.7) and to connect to (Figure 4.6) the neighboring net token instances living in the adjacent places. It relies on *intra-layer reading* for neighbors identification, and on *cross-layer reading* and *writing* for communications between net tokens.

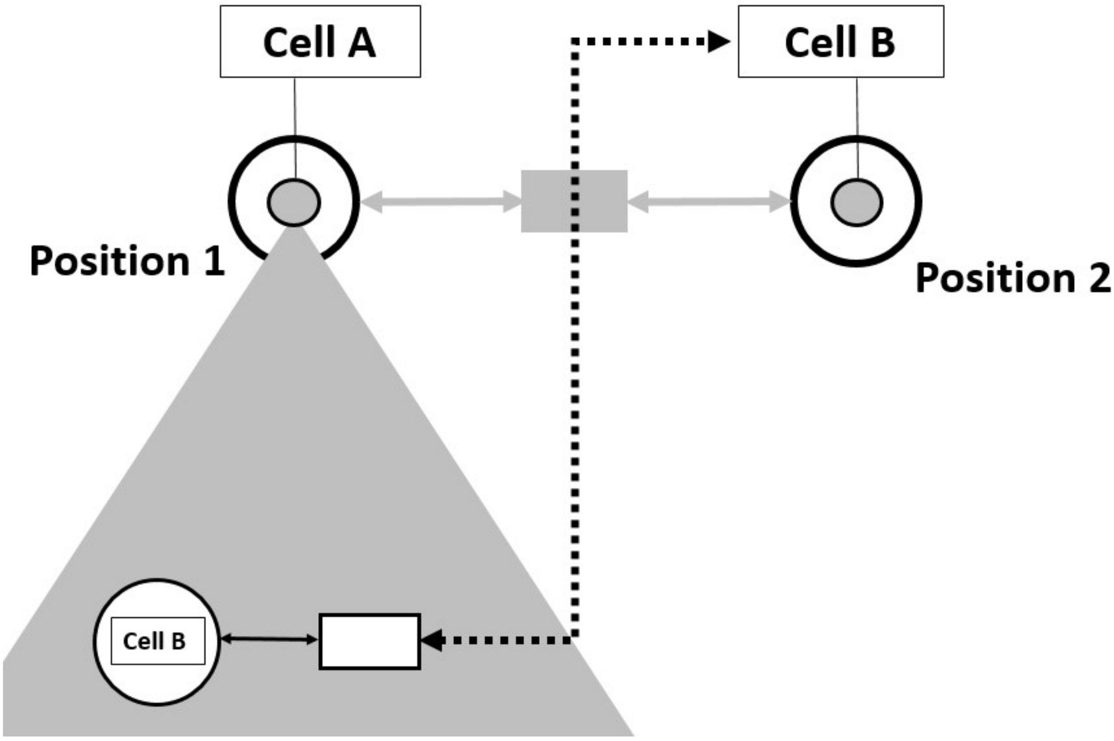


Figure 4.6. Neighbor detection

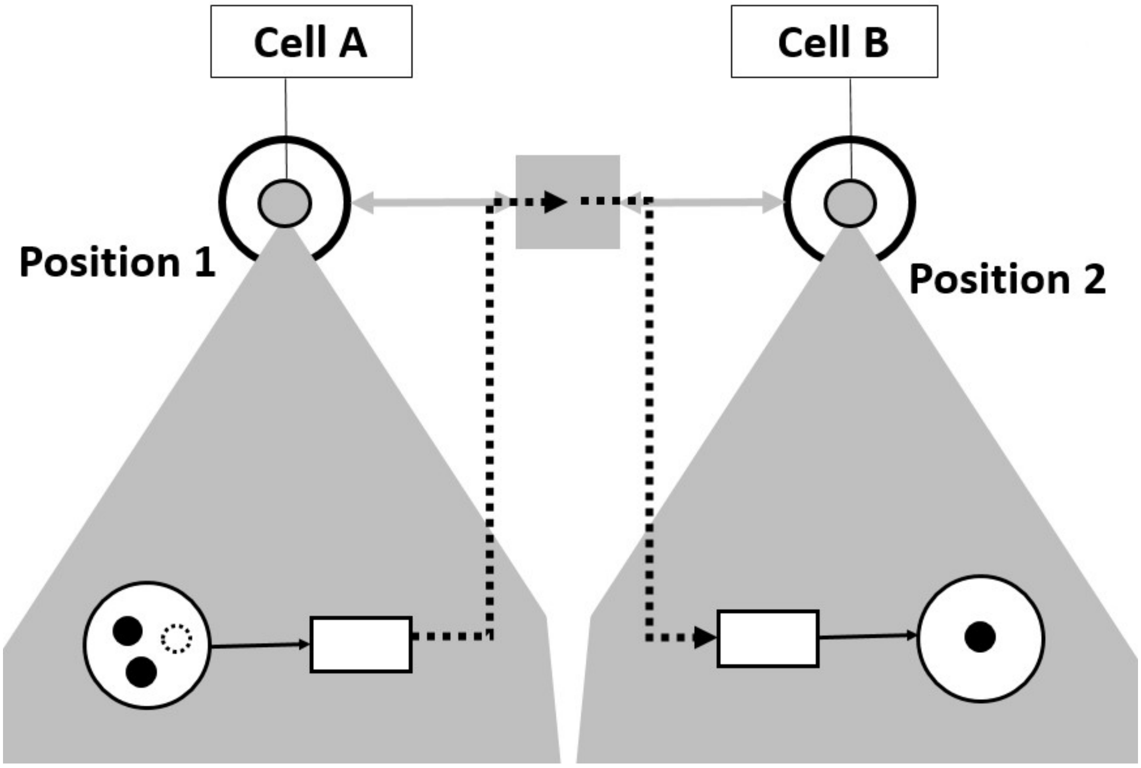


Figure 4.7. Neighbor communication

Cell movement models the atomic movement of a cell in the defined space with the transport of a net token from a place to another in the grid (Figure 4.8). It relies on *intra-layer writing*.

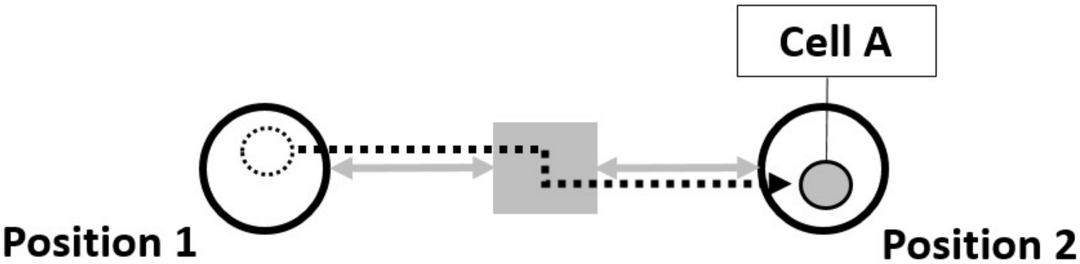


Figure 4.8. Cell movement

Molecular flow models the atomic movement of a biomolecule in the defined space with the transport of a colored token from a place to another in the grid (Figure 4.9). It relies on *intra-layer writing*.

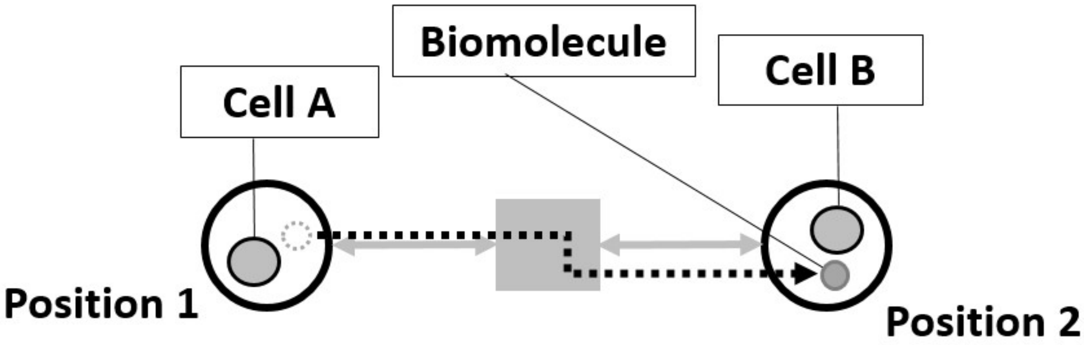


Figure 4.9. Molecular flow

Mitosis models the generation of two daughter cells from a single one with the consumption of a net token and the following instantiation of two copies of it (Figure 4.10). By *cross-layer reading*, a *checkpoint* ensures cell division starts after specific *markers* signal the completion of previous mitotic phases. The newly generated net tokens model daughter cells and occupy the starting place and one of the adjacent ones respectively. The choice of the latter can be random. Alternatively, contextual *rules* can affect the choice, including directionality over embryo axes, other neighbor cells, and gradients of biomolecules over the surrounding places.

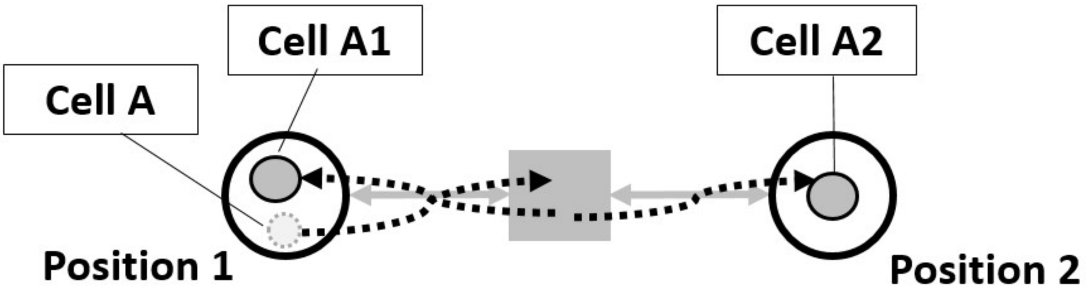


Figure 4.10. Mitosis

Apoptosis models the regulated death of a cell with the consumption of a net token from a place (Figure 4.11). By *cross-layer reading*, a *checkpoint* ensures the apoptotic process starts after specific regulations within the cell are in place and the respective *markers* arise.

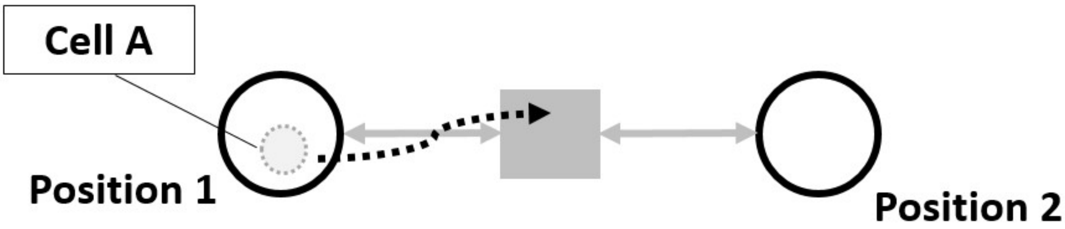


Figure 4.11. Apoptosis

Signal sensing models the passage of a signal, carried by a biomolecule, from outside to inside a cell. By *cross-layer writing*, the colored token modeling the signal flows into the net token modeling the cell when they both occupy the same place in the grid (Figure 4.12).

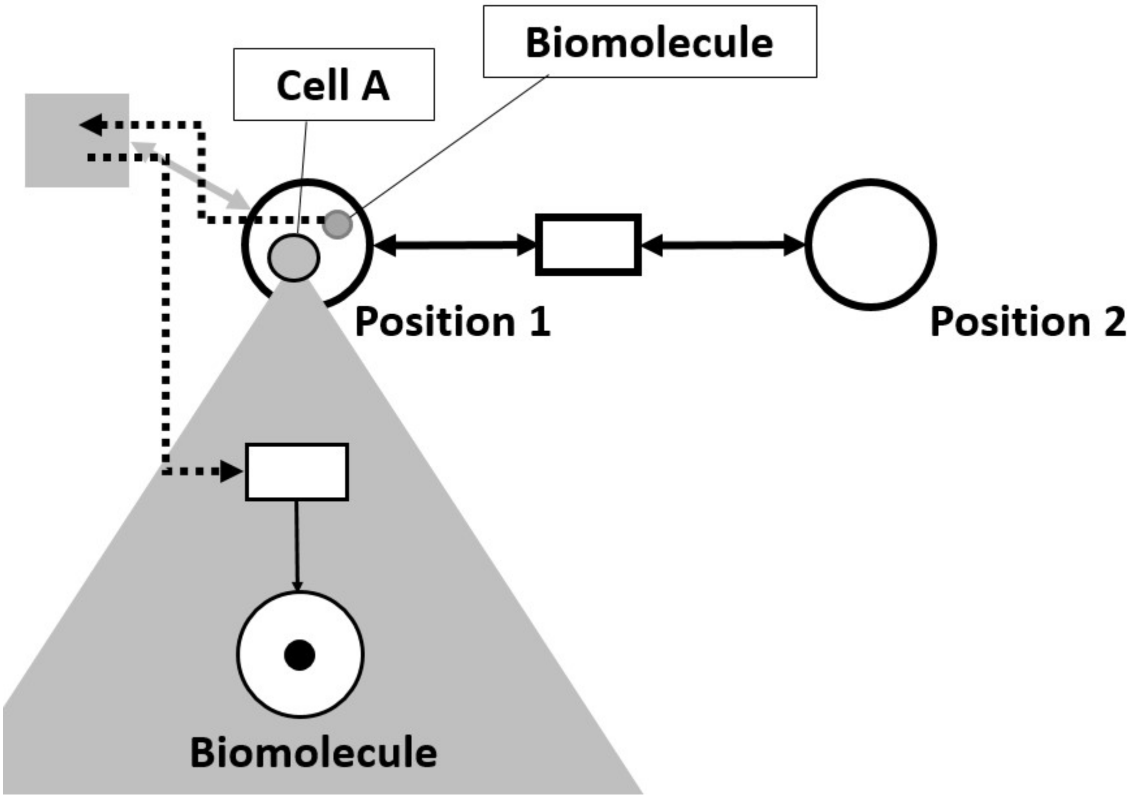


Figure 4.12. Signal sensing

Signal sending models the passage of a signal, carried by a biomolecule, from inside to outside a cell. By *cross-layer writing*, the colored token modeling the signal flows from the net token modeling the cell to its place in the grid (Figure 4.13).

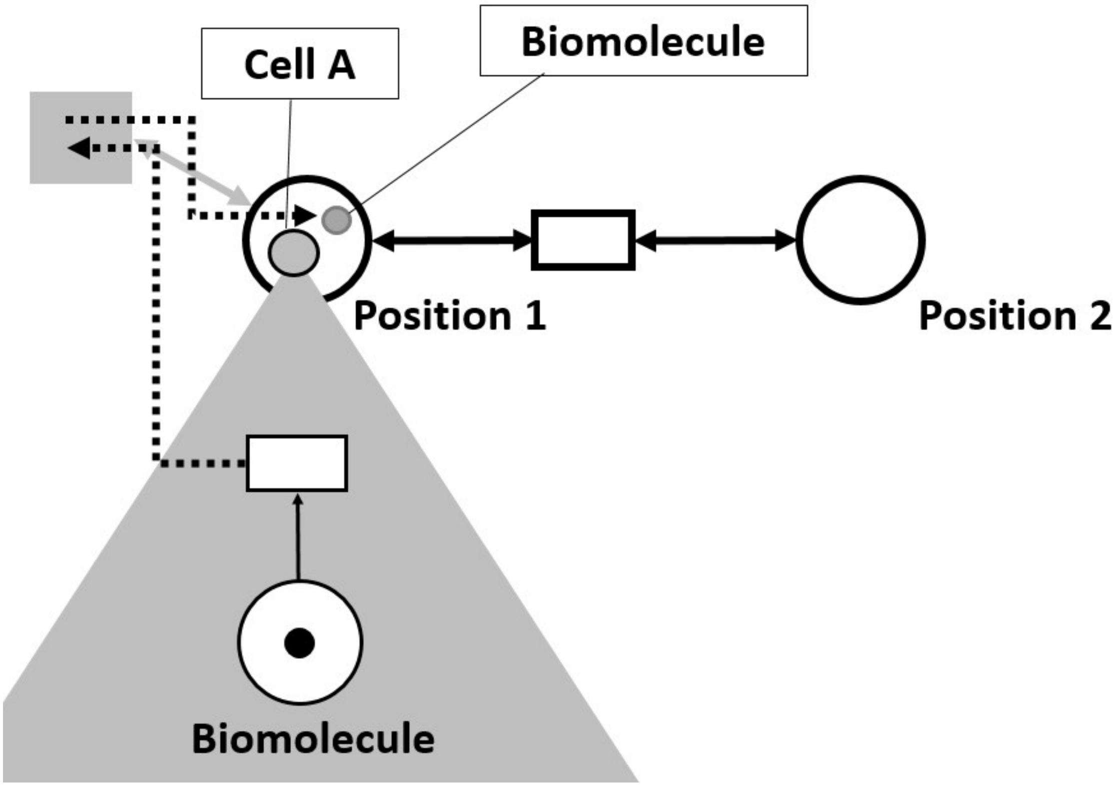


Figure 4.13. Signal sending

Differentiative Landscape

In the DL, the following relations between model and system elements hold:

- *Places* model cell states, inteded as functional identities or phenotypes;
- *Transitions* model passages from a cellular state to another one.
- *Net tokens* model cells;
- *Net architecture* models the landscape of differentiative trajectories underlying the ontogenetic process.

In the DL, model elements form the following basic building block, comprising *inductive* and *cell autonomous* ontogenetic mechanisms.

Differentiative step models the passage of cells from a state to another. By *cross-layer reading*, a *checkpoint* dynamically assesses the state of net tokens, and if they respond to the requirements, a state change takes place. Firing relies on *intra-layer writing* for transporting the net token to the place modeling the following state (Figure 4.14).

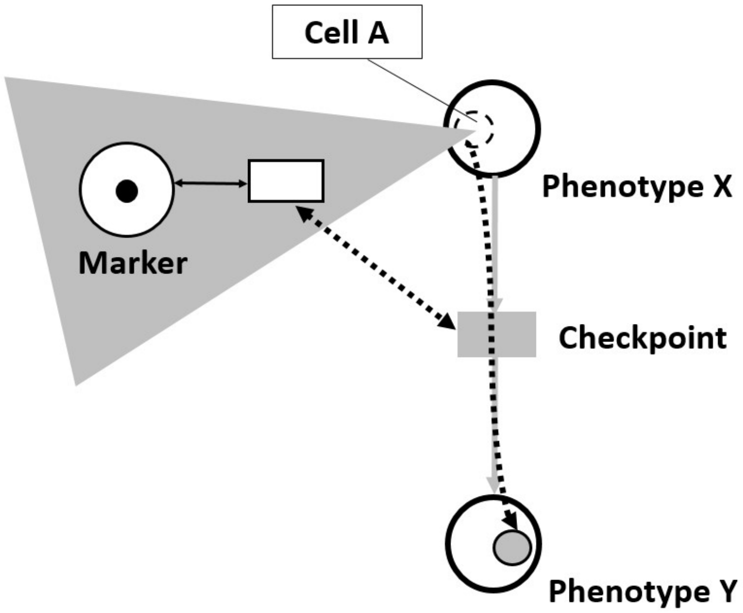


Figure 4.14. Basic building block from the DL: a general and abstract representations of the *differentiative step*. Checkpoint rules adapt to the specificities of the differentiative trajectory to be modeled.

Cells

In the Cells, the following relations between model and system elements hold:

- *Places* model biomolecules from all *-omics* and their possible states, including for example genes, mRNAs, ncRNAs, active and inactive proteins, and metabolites.
- *Transitions* model all kinds of biological processes, for example, transcription, translation, genetic and epigenetic regulation, post-translational modification, enzymatic catalysis, and protein degradation;
- *Black tokens* model biomolecules within the cell, whose identity changes depending on the place they live into;
- *Net architecture* models the scheme of relations between bioprocesses and the flow of resources among them.

Model elements form the following *building blocks*, covering basic *inductive* and *cell autonomous* ontogenetic mechanisms.

Transcription models the use of genetic information for producing protein-coding (mRNA) or non-coding transcripts. By *intra-layer reading*, a black token marking the presence of a gene allows, without being consumed, for the production of a variable number of black tokens in the places modeling the transcriptional products of that gene (Figure 4.15).

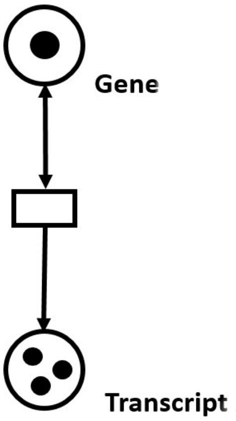


Figure 4.15. Transcription

Translation models the consumption of an mRNA for producing an aminoacidic chain, or protein. A black token in the place modeling the mRNA is consumed for producing one (or more) black tokens in the place modeling the protein product (Figure 4.16).

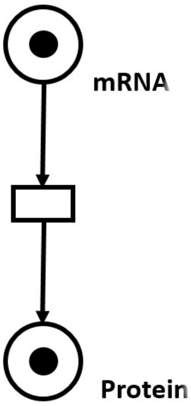


Figure 4.16. Translation

Enzymatic reaction models the modification of the state or structure of a biomolecule through the intervention of an enzyme, which may be the same or another biomolecule. By intra-layer reading, it checks for the presence of the active enzyme. After that, it consumes black tokens from the place modeling the substrates for the reaction and produces black tokens into the place modeling its products, following its stoichiometry. This general scheme can model diversity of reactions, for example, protein activation by post-translational modification, as well as metabolic cycles (Figure 4.17).

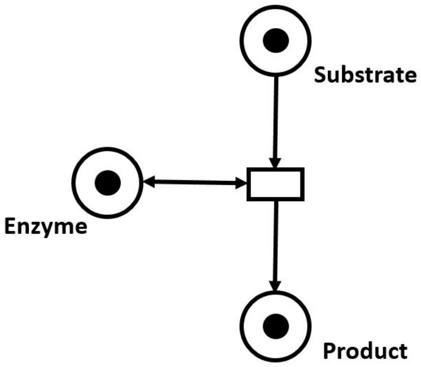


Figure 4.17. Enzymatic reaction

Gene regulation models the role of regulatory molecules in the modulation of gene expression. This block can attach to a transcription block, which will consume black tokens from a place modeling the regulator (for instance, a transcription factor) to switch on, off or modulate the process (Figures 4.19 and 4.18).

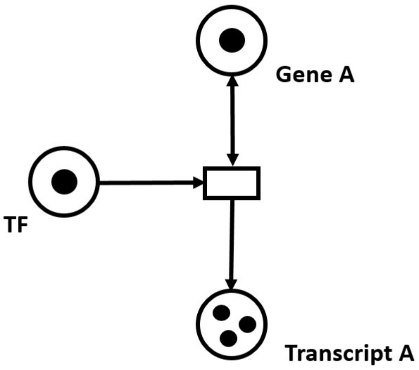


Figure 4.18. Activating gene regulation

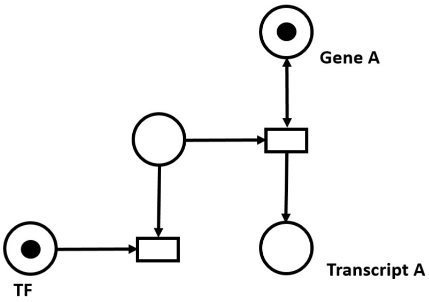


Figure 4.19. Inhibiting gene regulation

Post-transcriptional regulation models the role of molecules able to contribute to the subtle modulation of mRNA translation, such as miRNAs. This block can attach to the place modeling coding transcripts in a translation block, and, consuming some black tokens from the place modeling miRNAs, take away some black tokens modeling mRNAs, according to the specific stoichiometry, and produces black tokens into the place modeling mRNA with miRNAs attached (Figure 4.20).

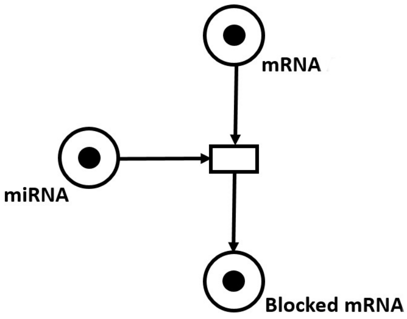


Figure 4.20. Post-transcriptional regulation

Functional modules for complex ontogenetic mechanisms

Higher-level *functional modules* for complex ontogenetic processes can be seen as combinations of building blocks. They encompass multiple system levels, providing holistic representations of complex ontogenetic phenomena. In our approach, all model levels contribute to support *functional modules*, relying on cross-layer communication mechanisms that ensure consistency in the resources and information flow across nets, and semantic coherence of the overall model.

At the aim to provide some applicative examples of the presented modeling approach, we provide some examples of complex ontogenetic mechanisms *functional modules* can model, and the basic building blocks they can rely on to do so. The Discussion and Conclusions section addresses the question of the role of the modelers, and when they stop to assemble and adapt pre-existing structures and start designing their model.

Migration waves can correspond to different biological mechanisms, all devising the active movement of cells, co-directed by other cells and environmental factors such as mechanical or chemical gradients. *Cell movement*, *neighbor sensing*, *molecular flow* and *signal sensing* building blocks combine to model this process. [how?]

Apoptotic waves involve a group of cells undergoing apoptosis in a regulated way, often inducing proliferation and migration in neighbor cells. *Molecular flow*, *Signal sensing* and *apoptosis* building blocks, over a set of adjacent places in the ISG underlie these processes.

Proliferative phases involve a selected population of cells undergoing mitotic processes in a regulated way, as a form of morphogenetic mechanism. *Signal sensing* and *mitosis* building blocks underlie this process, starting from places in the ISG and populating adjacent ones.

Patterning is the emergence of phenotype and architectural complexity from the local interactions between cells. It results from the combination of *inductive* mechanisms. For example, *hierarchical* signaling from a unique signal source can determine a chemical gradient over the architecture of receiver cells. In a distance-dependant way, cells receive a graded signal, having different effects at different concentrations. This can determine *per se* a *pattern* of different cell identities. Lateral signaling between neighbor receiver cells can affect the downstream effects of the signal.

4.4.2 Application example: VPC specification in C. Elegans

This section presents, as an application example of the modeling strategy developed for ontogenetic processes, the Vulval Precursor Cells (VPC) specification process in C. Elegans. The corresponding model relies on the *patterning* complex ontogenetic mechanism (see 4.4.1), introducing into the details the corresponding *functional modules*.

As extensively described in Riddle et al., 1997; Schmid and Hajnal, 2015; Shin and Reiner, 2018; Sternberg, 2005, VPC specification occurs between the L3 and L4 stages of larval development in C. Elegans. At this stage, each of six multi-potent stem cells, the Pn.p cells (Figure 4.21, acquire one of three fates (1°, 2° or 3° fate respectively). Fate determination in this phase guides the subsequent phases of organ development. For each Pn-p cell, different actors contribute to fate decision:

- the Anchor Cell (AC), residing in the adjacent developing uterus district;
- the hypodermis, residing below all Pn.p cells;
- the neighbor Pn.p cells.

Each of these actors can affect fate determination in each Pn.p cells, via different mechanisms. Intra-cellular regulation mechanisms mediate the effects of environmental signals via specific pathways.

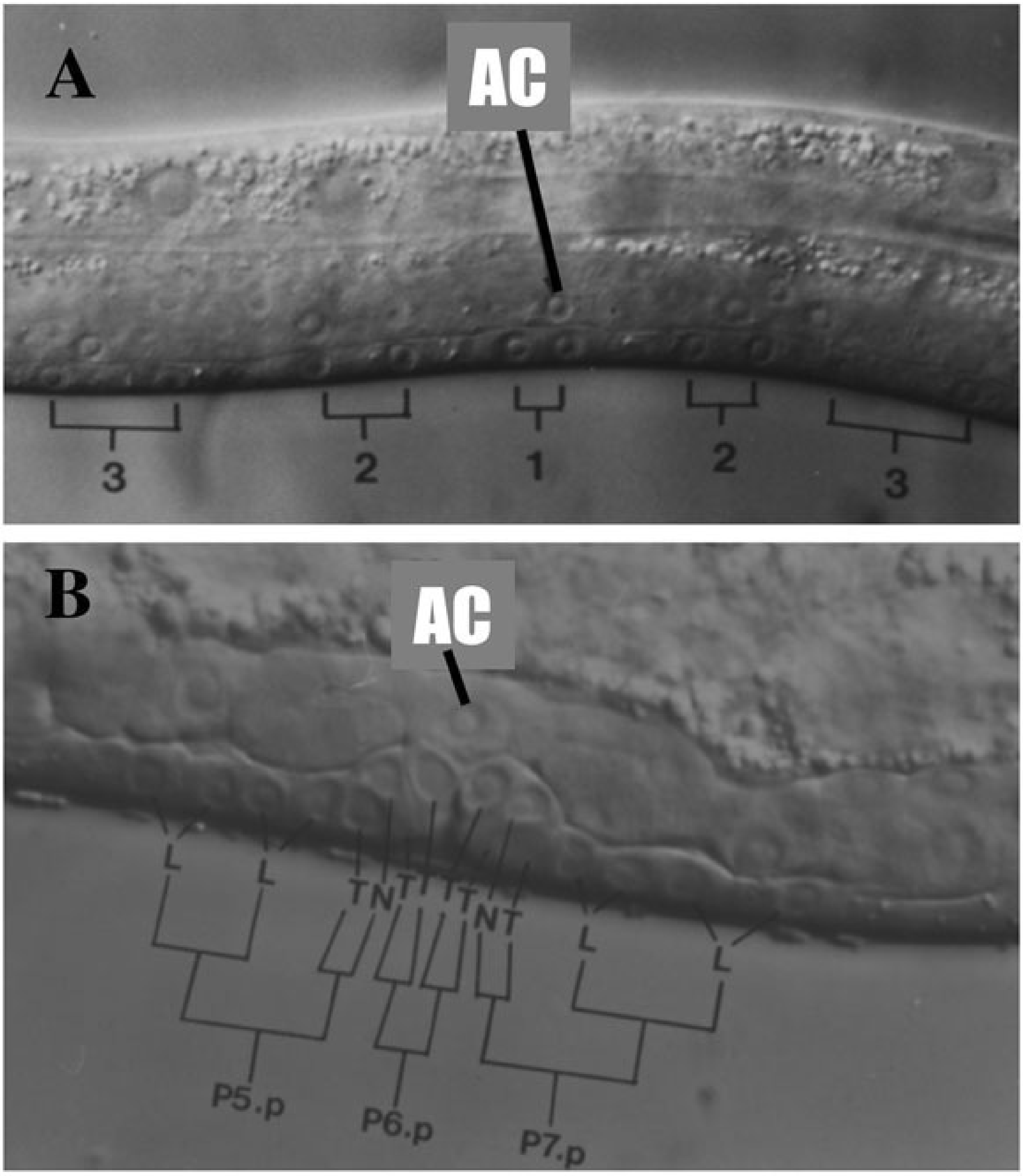


Figure 4.21. Left lateral views of VPC specification in C. Elegans larval hermaphrodites. **A**: the configuration from the L3 developmental stage: no signs of morphological differentiation. **B**: two rounds of mitosis over the cell lineage are shown. The 2° and 1° vulval precursors take part to longitudinal (L), transverse (T) or no division (AC: anchor cell; 3: 3° VPCs (P4.p and P8.p); 2: 2° VPCs (P5.p and P7.p); 1: 1° VPC).

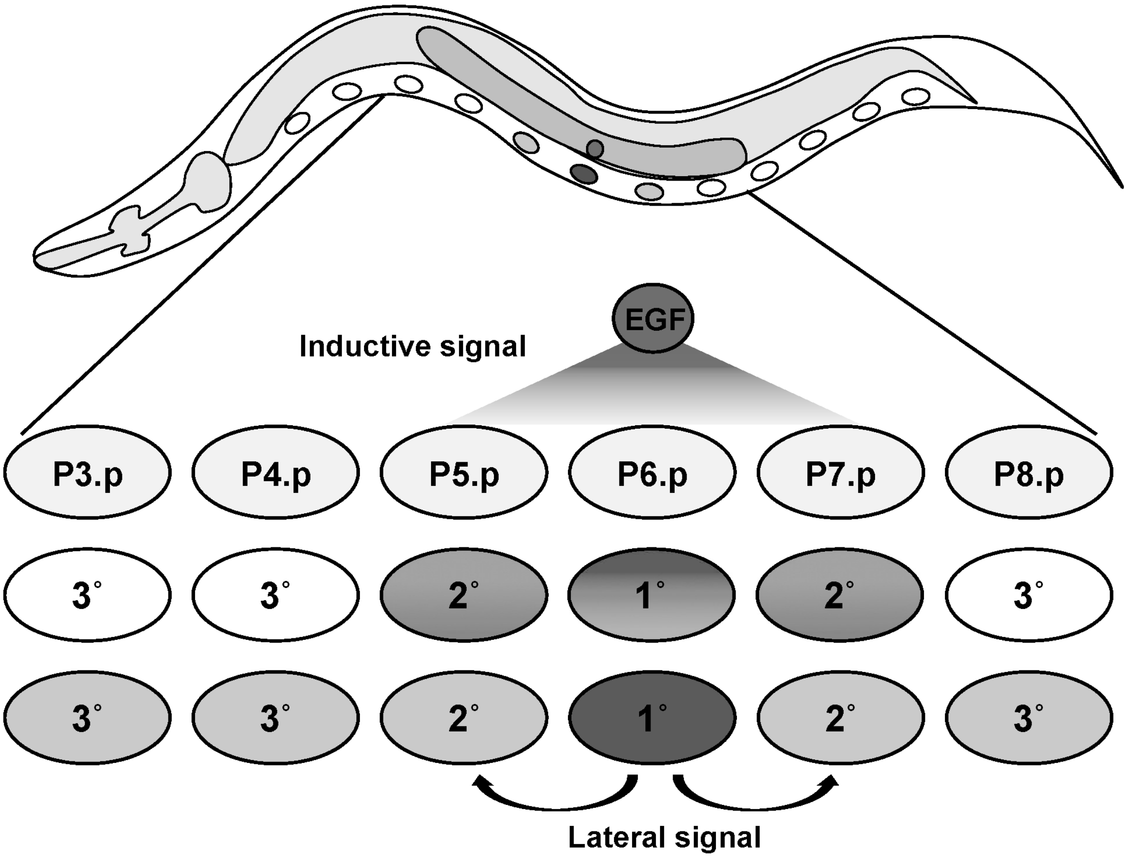


Figure 4.22. Overview of the C. Elegans VPC fate patterning. Pn.p cells are numbered P3.p through P8.p. P6.p, the closest to the Anchor Cell (AC), receives the highest level of LIN-3 inductive signal (via juxtacrine signaling) and assumes 1° fate. P5.p and P7.p receive lower levels of inductive signal (via paracrine signaling) and lateral Notch signal from the P6.p, yielding to 2° fate. P3.p, P4.p, and P8.p, due to the lack of sufficient signal levels, adopt nonvulval fates. Adapted from Shin and Reiner, 2018.

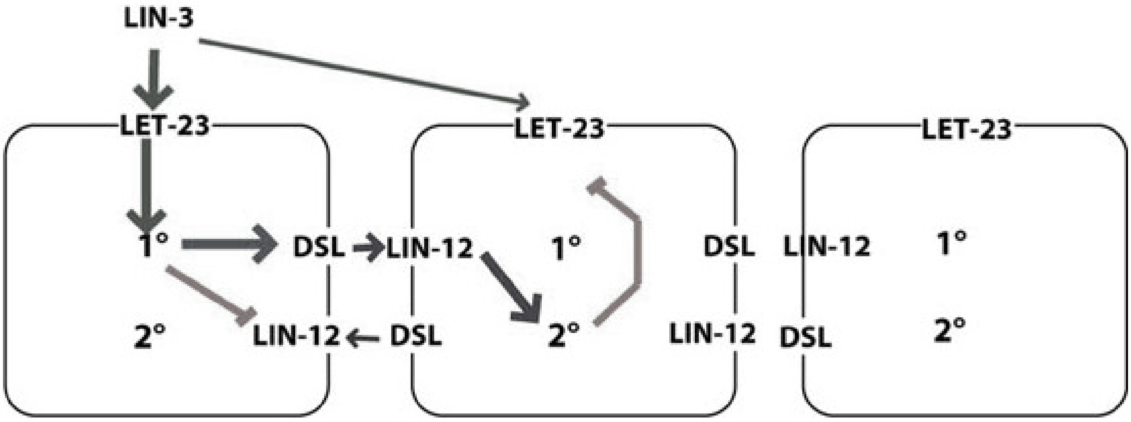


Figure 4.23. Critical aspects of the 1°-2° VPC patterning mechanism. LIN-3 from the anchor cell acts in a graded fashion with P6.p receiving more signal than P7.p. LET-23 activation promotes the 1° fate and production of DSL ligands for LIN-12, and inhibits LIN-12 protein levels. LIN-12 activation promotes the 2° fate and inhibits the response to LET-23 activation. Adapted from Sternberg, P.W., Vulval development (June 25, 2005), WormBook, ed. The C. Elegans Research Community, WormBook, doi/10.1895/wormbook.1.6.1, http://www.wormbook.org. © 2005 Paul W. Sternberg.

As in Figure 4.23, in the physiological case, the AC sends out a LIN-3 (EGF-like) signal reaching Pn.p with distance-dependent intensity. The closest cell to the AC, P6.p, engages in LIN-3/LET-23 based juxtacrine signaling, while its neighbors, P5.p and P7.p cells, receive paracrine signaling, mediated by the soluble isoform of LIN-3. The signal does not reach P3.p, P4.p, P7.p and P8.p, the farthest cells. The hypodermis (hyp7) sends uniformly strong paracrine LIN-3 signals to all Pn.p cells. Each of the Pn.p cells can engage in mutual juxtacrine lateral signaling via trans-membrane DSL/LIN-12 (DSL/Notch-like) signaling.

At the intracellular level, intense LIN-3 signaling induces the 1° fate in P6.p, via the activation of a LET-23-mediated RAS/MAPK signaling pathway. The predominance of the 1° fate becomes evident by high concentrations of the active form of MPK1, which activates strong DSL lateral signaling to the neighbors, causing them to switch off the 1° fates traits induced by LIN-3 paracrine signals from the AC, activating 2° fate traits instead, corresponding to high concentrations of the active LIN12 protein. P3.p, P4.p, and P8.p cells do not receive any LIN-3 other than that from the hypodermis, thus causing them to undergo the 3° fate. In a non-physiological case, corresponding to the ablation of the AC, no LIN-3 inductive signal is present: all cells undergo the 3° fate.

The combinations of these signals, considering the states different cells can assume, results in a pattern of six cells, each one with a fate among three: primary, secondary or tertiary.

Patterning as a functional module

This model implementation recapitulates the complex ontogenetic mechanism defined as Patterning in Section 4.4.1. This functional module combines building blocks to model different mechanisms. Starting from the top level in the model, the ISG models different cell-cell communication mechanisms.

Juxtacrine signaling relies on *neighbor detection* and *neighbor communication*. This mechanism links the AC with the P6.p cell position, and the Cell instance living in it, as well as each Pn.p cell positions and Cell model instances with its neighbors. While the communication from the AC to P6.p is unilateral, between each pair of neighbor two signaling ways exist, since communication is bi-lateral.

Paracrine signaling relies on *signal sending*, *molecular flow* and *signal sensing*. This mechanism links the AC with P5.p and P7.p cells and represent a unilateral communication mechanism.

The DL model represents with places the possible states of the cells along the developmental step from L3 to L4.

Fate determination leverages one *differentiative trajectory* building block for each possible fate for Pn.p cells. Each trajectory relies on a *checkpoint* with different rules. All checkpoints track dynamically the marking evolution of Cell models respect to the places modeling active MPK1 and active LIN12. A *classification* module labels cells basing on the two resulting vectors of sampled values. After classification, each Cell model in the DL moves to the place corresponding to the acquired state, via a *cell movement* building block. The model is currently trained, with a supervised mechanism, on a dataset generated from previous simulation runs. Classification occurs at the System Net level during simulations. The simulation outputs the result of the six classifications: an ordered array of six labels, recapitulating the fate decision over each Pn.p cell. The Cell model represents with the low-level PNs formalism *cell autonomous* mechanisms, except for it communicates with the ISG and the DL.

Signaling pathways and their mutual interactions rely on *signal sensing* and *enzymatic reation.* Each actor involved in the pathway also has *gene regulation* and *translation* building blocks. In particular, [egf-like, dsl/notch like]

Fate determinants correspond to places, which the *checkpoint* building blocks from the DL access and use for classifying the cell into one out of the three places.

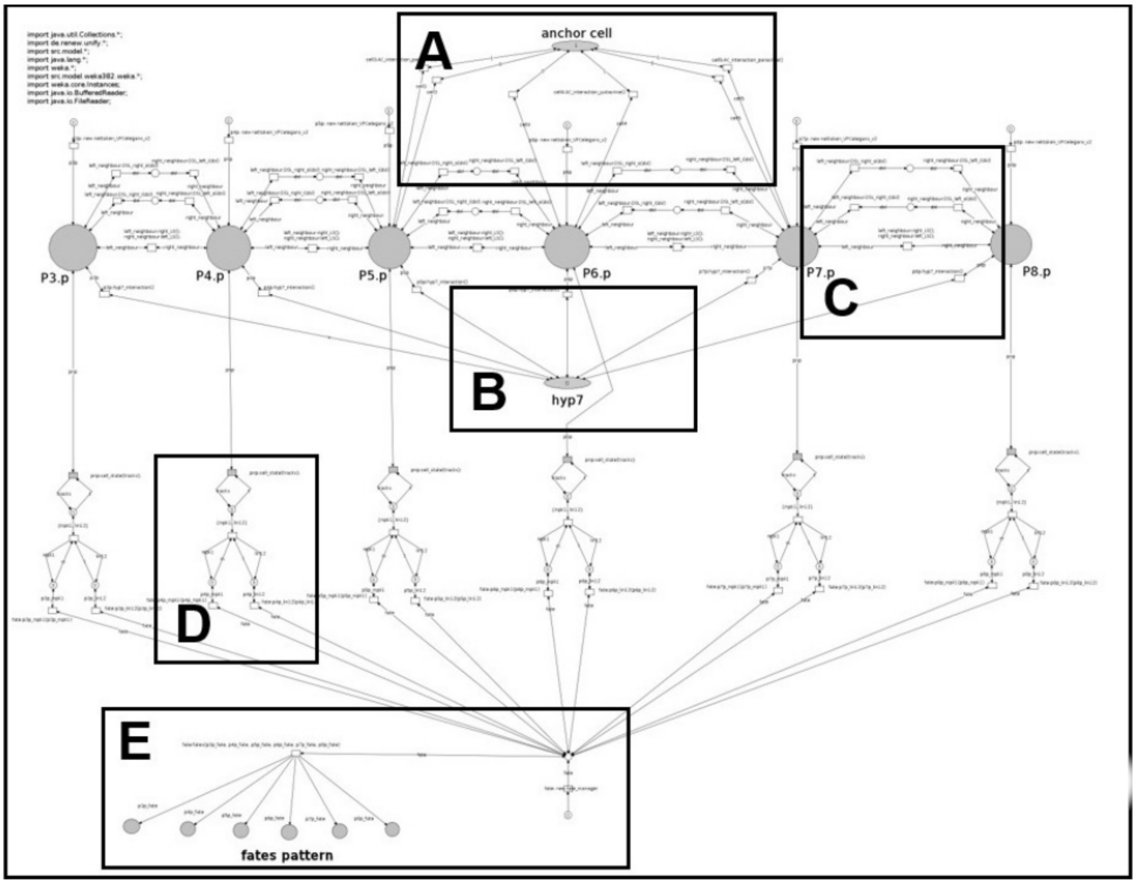
Model construction

The model aims to perform predictions over the fate decision processes for each cell, predicting, in the end, the fates pattern formation in the single developmental stage between L3 and L4. For modeling *autonomous* and *inductive* cellular mechanisms, this model integrates the one presented in Bonzanni, Krepska, et al., 2009, recapitulating the available information about the process. Information about the organization of cells forms a scheme of relative positions, which permits to build a bi-dimensional spatiality model, where it is possible to express proximity relations and graded discrete distances. Recapitulation of possible cell fate specifications draw the differentiative trajectories in the landscape model. Checkpoints assign cell fates dynamically, classifying cells according to their evolving markings during execution.

Model implementation

The resulting model comprises three views over the system.

Interactive Spatial Grid A bi-dimensional ISG model mediates juxtacrine and paracrine signaling by the AC, lateral signaling between Pn.p cells, and paracrine signals from hyp7. Six places represent positions where Pn.p cells are supposed to live, in their mutual proximity relations. One place hosts the AC in the form of a colored token, which can be either present or absent to this level of abstraction. Another place hosts hyp7 in the form of a black token.



```
import java.util.Collections.*;
import de.renew.unify.*;
import src.model.*;
import java.lang.*;
import weka.*;
import src.model.weka382.weka.*;
import weka.core.Instances;
import java.io.BufferedReader;
import java.io.FileReader;
```

Figure 4.24. In this implementation, the ISG model and the DL model combine, representing in two dimensions the relative positions of the AC and the Pn.p cells, and the states Pn.p cells can assume in terms of fate determination. **A**: The place modeling the AC links to P5.p, P6.p and P7.p positions in the grid via modules implementing cell-cell communication mechanisms, devising *neighbor detection*, *molecular flow* and *signal sensing* building blocks. **B**: the place modeling hyp7 leverages *neighbor detection* and *signal sensing* to communicate with all Pn.p positions. **C**: lateral signaling between neighbor cells rely on *neighbor detection*, *neighbor communication* and *molecular flow* building blocks. **D**: for each Pn.p position in the grid, the *checkpoint* building blocks sample, via cross-level communication, the marking evolution of the *fate determinants* from Cell models living there. **E**: After *checkpoints* classify each Cell instance, the results of the classification form the *pattern*, highlighting a new state out of the landscape of possible differentiative state for each of them.

Differentiative Landscape A DL model for the developmental stage from L3 to L4 comprises the state of Pn.p cell, 1°, 2°, and 3° fates respectively. This model also represents the trajectories from the Pn.p state to each of them. Each trajectory has a checkpoint, classifying the cell by the time evolution of its levels of active MPK1 and LIN12 as having one out of the three fates. In this implementation, the ISG and the DL models coexist in the same net (Figure 4.24).

In particular, in this implementation, the *checkpoint* has a model *per se*, which takes the name of Fates Manager (Figure 4.25). In this net, the classifiers are operated by structures for dynamically storing reading vectors for the classification, so that it can occur at runtime over the current time windows of markings from the cells.

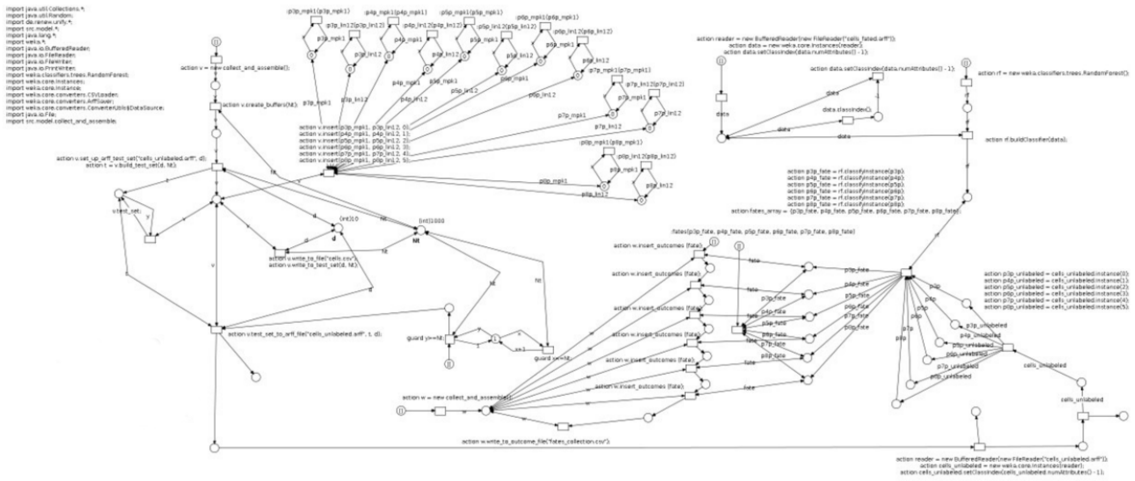

import java.util.Collections.*;
import java.util.Random;
import de.renew.unify.*;
import src.model.*;
import java.lang.*;
import weka.*;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;
import weka.classifiers.trees.RandomForest;
import weka.core.Instances;
import weka.core.Instance;
import weka.core.converters.CSVLoader;
import weka.core.converters.ArffSaver;
import weka.core.converters.ConverterUtils$DataSource;
import java.io.File;
import src.model.collect_and_assemble;
action v = new collect_and_assemble();
action v.create_buffers(Nt);
action v.set_up_arff_test_set("cells_unlabeled.arff", d);
action t = v.build_test_set(d, Nt);
v.test_set;
:p3p_mpk1(p3p_mpk1)
:p3p_lin12(p3p_lin12)
:p4p_mpk1(p4p_mpk1)
:p4p_lin12(p4p_lin12)
:p5p_mpk1(p5p_mpk1)
:p5p_lin12(p5p_lin12)
:p6p_mpk1(p6p_mpk1)
:p6p_lin12(p6p_lin12)
:p7p_mpk1(p7p_mpk1)
:p7p_lin12(p7p_lin12)
:p8p_mpk1(p8p_mpk1)
:p8p_lin12(p8p_lin12)
action v.insert(p3p_mpk1, p3p_lin12, 0);
action v.insert(p4p_mpk1, p4p_lin12, 1);
action v.insert(p5p_mpk1, p5p_lin12, 2);
action v.insert(p6p_mpk1, p6p_lin12, 3);
action v.insert(p7p_mpk1, p7p_lin12, 4);
action v.insert(p8p_mpk1, p8p_lin12, 5);
(int)10
(int)1000
action v.write_to_file("cells.csv");
action v.write_to_test_set(d, Nt);
action v.test_set_to_arff_file("cells_unlabeled.arff", t, d);
guard y!=Nt;
guard x==Nt;
action w = new collect_and_assemble();
action w.insert_outcomes (fate);
action w.write_to_outcome_file("fates_collection.csv");
action reader = new BufferedReader(new FileReader("cells_fated.arff"));
action data = new weka.core.Instances(reader);
action data.setClassIndex(data.numAttributes() - 1);
action data.setClassIndex(data.numAttributes() - 1);
data.classIndex();
action rf = new weka.classifiers.trees.RandomForest();
action rf.buildClassifier(data);
action p3p_fate = rf.classifyInstance(p3p);
action p4p_fate = rf.classifyInstance(p4p);
action p5p_fate = rf.classifyInstance(p5p);
action p6p_fate = rf.classifyInstance(p6p);
action p7p_fate = rf.classifyInstance(p7p);
action p8p_fate = rf.classifyInstance(p8p);
action fates_array = {p3p_fate, p4p_fate, p5p_fate, p6p_fate, p7p_fate, p8p_fate};
:fates(p3p_fate, p4p_fate, p5p_fate, p6p_fate, p7p_fate, p8p_fate)
action p3p_unlabeled = cells_unlabeled.instance(0);
action p4p_unlabeled = cells_unlabeled.instance(1);
action p5p_unlabeled = cells_unlabeled.instance(2);
action p6p_unlabeled = cells_unlabeled.instance(3);
action p7p_unlabeled = cells_unlabeled.instance(4);
action p8p_unlabeled = cells_unlabeled.instance(5);
cells_unlabeled
action reader = new BufferedReader(new FileReader("cells_unlabeled.arff"));
action cells_unlabeled = new weka.core.Instances(reader);
action cells_unlabeled.setClassIndex(cells_unlabeled.numAttributes() - 1);

Figure 4.25. In this implementation, an entire net dedicates to managing the rules behind *checkpoint* building blocks. Thanks to cross-layer communication mechanisms, this net dynamically reads, and stores evolving *fate determinants* MPK1 and LIN12 markings from each Cell instance, and leverages a Weka-based classification model for labeling them with a specific fate.

Cells A Cell model represents cell-autonomous mechanisms and signaling pathways communicating with other cells via the ISG, such as the MPK signaling cascade, and the DSL/LIN12 pathway. For each actor from signaling pathways gene regulation and protein degradation processes find a representation in the model (Figure 4.26).

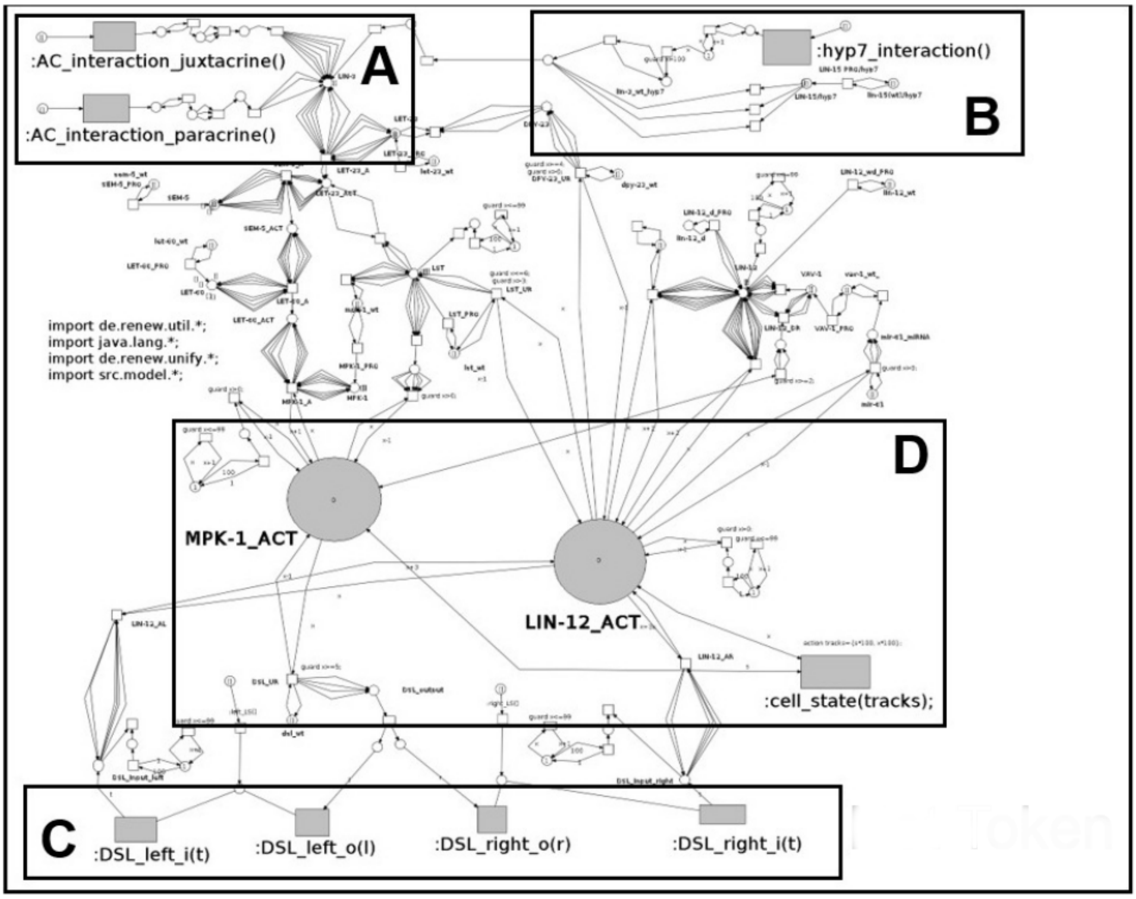


Figure 4.26. The Cell model, recapitulating cell-autonomous mechanisms in Pn.p cells, and their communications within each other and with the environment. The MAPK and DSL/NOTCH signaling pathways interact resulting in different levels of active MPK1 and active LIN12 over simulation time. **A, B**: *signal sensing* building blocks mediate the AC-and hyp7-derived signaling mechanisms. **C**: *neighbor communication* building blocks, one for each side of the Cell, handle the lateral communication mechanisms between neighbor Cell instances. **D**: the *checkpoint* module samples evolving markings in the places modeling the *fate determinants* active MPK1 and active LIN12.

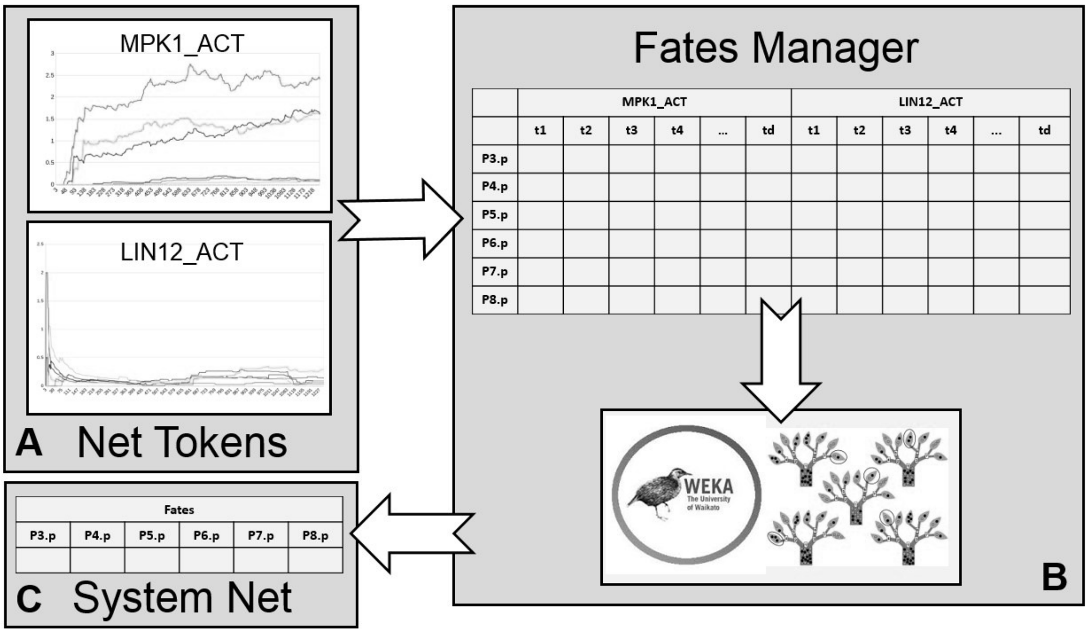
The Cell model lives as an instance in Pn.p places from the ISG and the Pn.p state in the DL model.

Experimental design

In the model, different initial markings represent different experimental conditions, chosen by the scheme proposed in Bonzanni, Krepska, et al., 2009. In particular, we simulate the physiological case and the AC ablated mutant. The latter devises the absence of the AC and the following functional activations, resulting in all cells acquiring the tertiary fate. During simulations, the final pattern of Pn.p cells fates constitutes the outcome. Such fates emerge from the evolution of marking in places modeling active MPK1 and active LIN12 respectively in each cell. Figure 4.27 shows two sample tracks the simulation automatically generates. Also, it depicts the fate pattern generation process, including the classification of these tracks into one of the three possible fates. This last step is based on the integration of the Java-based machine learning library Weka into the model.

Simulations

Model design and simulation rely on Renew Cabac, Haustermann, et al., 2016, a Java-based modeling environment for NWNs. Among the possible modes, the stochastic simulation mode allows representing stochastic effects by randomizing the firing order of enabled transitions. The classification module for the checkpoint in the DL relies on a custom Java class that integrates a RandomForest classifier based on Weka Smith and Frank, 2016, a Java-based library for machine learning.



	MPK1_ACT						LIN12_ACT					
	t1	t2	t3	t4	...	td	t1	t2	t3	t4	...	td
P3.p												
P4.p												
P5.p												
P6.p												
P7.p												
P8.p												

Fates					
P3.p	P4.p	P5.p	P6.p	P7.p	P8.p

Figure 4.27. A scheme of the classification process which in this implementation the *checkpoint* building block relies on for labeling cells with a fate. The classification process relies on a Weka random forest classifier, and its training makes use of previous simulation tracks, and it labels Cells with one out of the three possible fates basing on the evolving markings of active MPK1 and active LIN12, sampled over simulation time.

For each experimental condition, 100 simulations were performed, according to what authors in Ritter et al., 2011 recommend. Table 4.1 recapitulates the initial conditions tested, the relative expected pattern in the relative simulation outcome, and the accuracy scores per cell in each generated pattern. Simulation outcome is an array of six ordered fate values, one for each Pn.p cell, as in column *Pn.p pattern.*

experiment	Pn.p pattern	accuracies	refs
wt	3 3 2 1 2 3	100% 100% 100% 94% 100% 100%	(a)
AC ablated	3 3 3 3 3 3	100% 100% 100% 100% 100% 100%	(b)

Table 4.1. Each experimental condition corresponds to an expected Pn.p fates pattern. Accuracy scores refer to the cell-by-cell classification performance by the model, compared to the expected fate. Reported results refer to 100 simulation runs. References: (a)Sulston and Horvitz, 1977; (b)Kimble, 1981.

4.5 A modeling approach to synthetic biology

This section illustrates a potential role for the presented modeling approach for synthetic biology. For a more thorough exploration of the use of computational models in this field, and a detailed description of the presented use case, see Bardini, Di Carlo, et al., 2018. The domain of synthetic biology challenges the presented approach in at least two ways. On one side, it leverage models for design processes, where predictive power allows for reliable implementations. On the other side, synthetic systems may have operable components, for example, inducible genetic switches, or controllable culture conditions. The capability for inducing a state change in the system corresponds to a different interpretation of the states the system can assume. In fact, in these models, state changes need to be imposable in a top-down manner, reflecting the core paradigm shift in synthetic biology.

There is no single definition for the domain of synthetic biology. It generally concerns the implementation of an artificial function through modification or construction of a biological structure Cameron et al., 2014. In other words, what defines synthetic biology is the application of a top-down engineering approach to biological systems Benso, Di Carlo, Politano, Savino, and Bucci, 2014.

When engineering the genome of a complex biological system, it is necessary to precisely and reliably map the quantitative relation between the genetic information to be manipulated and its structural-functional correlates. This map is relevant at the very least for the downstream informational flow specific to the synthetic structure. It would be better to leverage information from the whole functional context for the modification. In other words, it is necessary to consider synthetic biological systems under the perspective of systems biology. In particular, it is mandatory to take into account, for each system level of interest, quantitative aspects, non-linearities, and stochasticity, as well as the way spatiality mediates regulation mechanisms in the system Bardini, Politano, et al., 2017a.

In the design of biological systems, the designer can either modify existing genetic information or build up complex artificial constructs to embed in an existing organism. The latter requires the definition and use of biological sub-parts for the construct to take shape and to yield the desired behavior Purnick and Weiss, 2009. This strategy, based on the use of functional building blocks from different system levels, highlights the necessity of a systemic perspective: each building block and their combination need to take into account the inherent complexity of the system.

The scope of this section is to illustrate how the presented modeling strategy can be employed to leverage the holistic understanding of systems biology to combine it with the top-down approach of synthetic biology.

4.5.1 Case-study: synthetic cell-cell communication

The presented case-study, based on W.-D. Wang et al., 2008, concerns the artificial culture of mammalian cells expressing the genes of a cell-cell communication genetic construct. The system includes two different types of cells, each one carrying a part of the construct, and engaging in a communication mechanism based on these parts. Also, the system comprises the cell culture medium, providing nourishment and possibly signals to the cells. Figure 4.28 recapitulates the functioning of such genetic circuit. The culture medium can function as an additional regulative layer for the cells, taking part in the artificial set-up for building the synthetic system.

Intercellular nitric oxide (NO)-based signaling mechanisms enable communication between sender and receiver cells respectively. Before that, the synthetic process of NO takes place, in sender cells, after induction. The NO-mediated signal reaches the receiver cells and acts inducing the expression of an EGFP reporter. More specifically, sender cells express NO synthetase (NOS) in a TeT-on-inducible way. Integrating with the c-fos promoter, NOS catalyzes the production of NO. Fos can activate, in receiver cells, an EGFP-based gene reporter. The NO signal activates soluble guanylyl cyclase, producing cytosolic GMP from GTP, which activates the reporter gene through fos. In this way, the receiver cell produces the EGFP reporter.

This section aims to show how the presented modeling strategy can cover the different levels of the biological and artificial systems involved in this design. In this perspective, the model needs to include the level of cells and that of the culture medium. In order to evaluate cell-cell communication, their spatial organization needs to find proper representation.

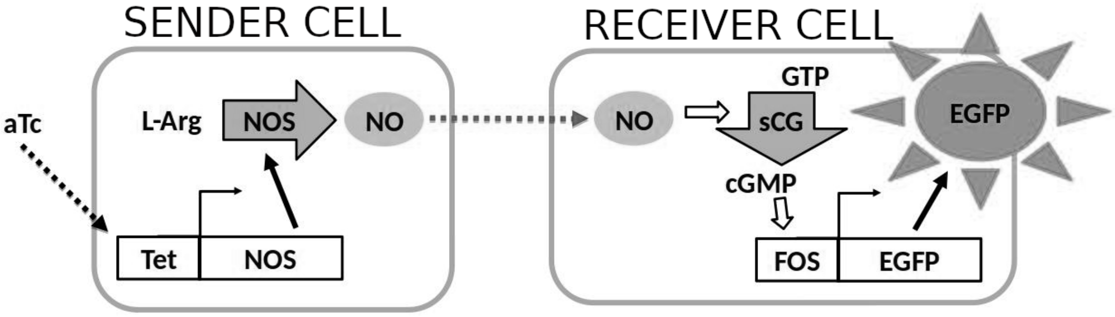


Figure 4.28. The synthetic circuit of interest involves different cell recipients, defined sender and receiver cells respectively. Sender cells send a NO signal after an aTc-inducible transcriptional activation, an receiver cells produce EGFP following the subsequent activation of a NO-mediated activation. The co-culture of sender and receiver cells promotes the activation of the NO signaling-based EGFP production. Adapted from Bardini, Politano, et al., 2018

Model design

The model represents three system levels:

- the culture medium;
- the cell aggregates;
- the mechanisms within cells.

Following the presented modeling strategy, three nets compose the model.

- the ISG models the volume cells occupy within the culture medium, and the spatial relations between them. It also mediates the NO-based signaling mechanisms, and the administration of signals to cells via the medium, as well as the detection of fluorescent cells in the cell aggregate (Figure 4.29);
- the DL model, which in this case has more of a *States* Landscape acceptation, comprises the identities and states of cells in the cultured system: sender and receiver, but also fluorescent and non-fluorescent. Another higher-level States Landscape operates the administration of the stimuli to the cell culture (Figure 4.30).
- the Cell models represent the artificial circuitry inserted in sender and receiver cells respectively; also, it possibly comprises all relevant cell *autonomous* mechanisms the designer may want to consider (Figure 4.31).

This type of model can be useful as a starting point to include more of the complexity from the modified system and to analyze the culture process. In the present form, the model only covers the synthetic circuit to be embedded in cells, and the cellular interaction it makes possible. For showing how the presented strategy can help this kind of process, in the following paragraphs, the three models and their building blocks are presented.

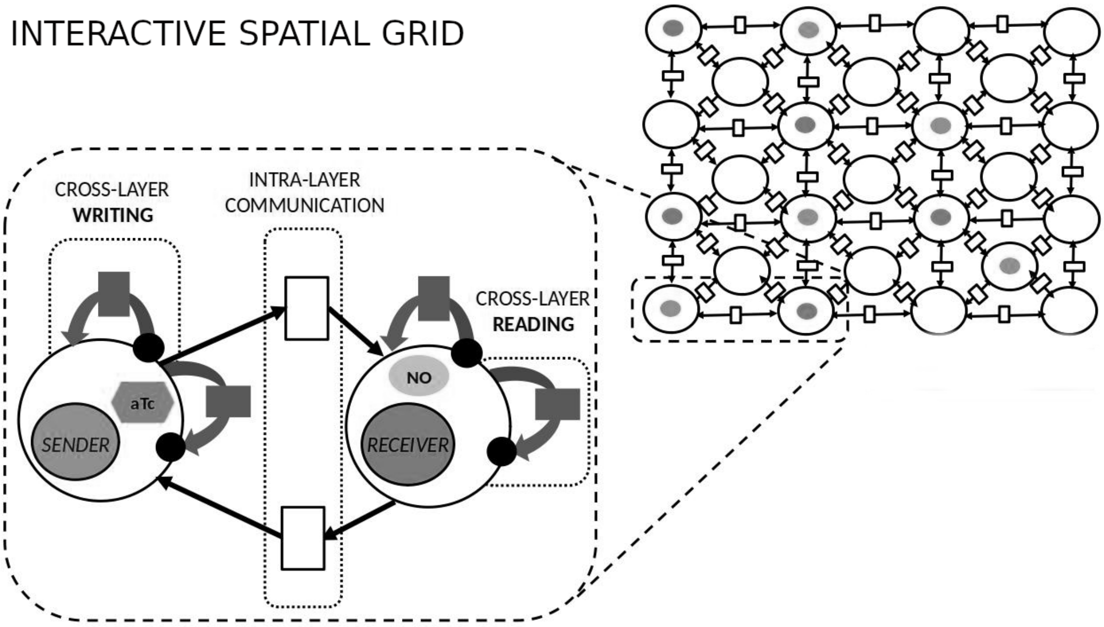


Figure 4.29. The Interactive Spatial Grid hosts both sender and receiver Cell instances. In the ISG, *molecular flow* building blocks model the diffucion of NO and aTc signals in the extracellular space. *Signal sending* and *signal sensing* building blocks model the generation and reception of aTc and NO signals. Adapted from Bardini, Politano, et al., 2018

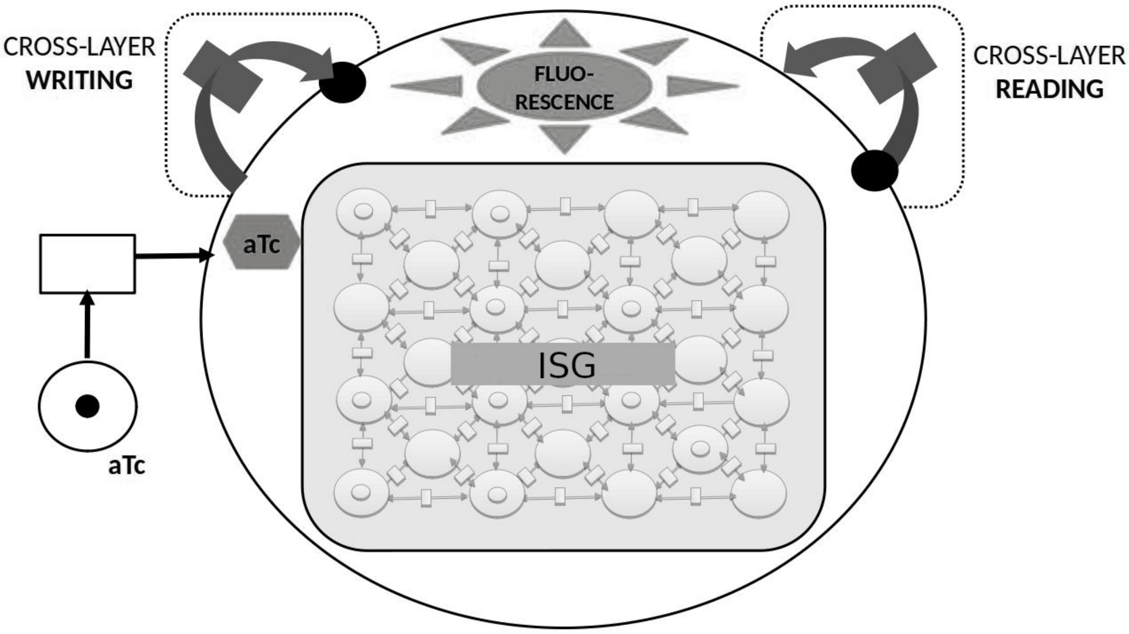


Figure 4.30. The culture environment functions as a control system for the cultured cells. Via *molecular flow* and *signal sensing* building blocks it can target the cell culture with the aTc signal. A checkpoint building block detects the presence of EGFP-based fluorescent signal from the lower level. Adapted from Bardini, Politano, et al., 2018

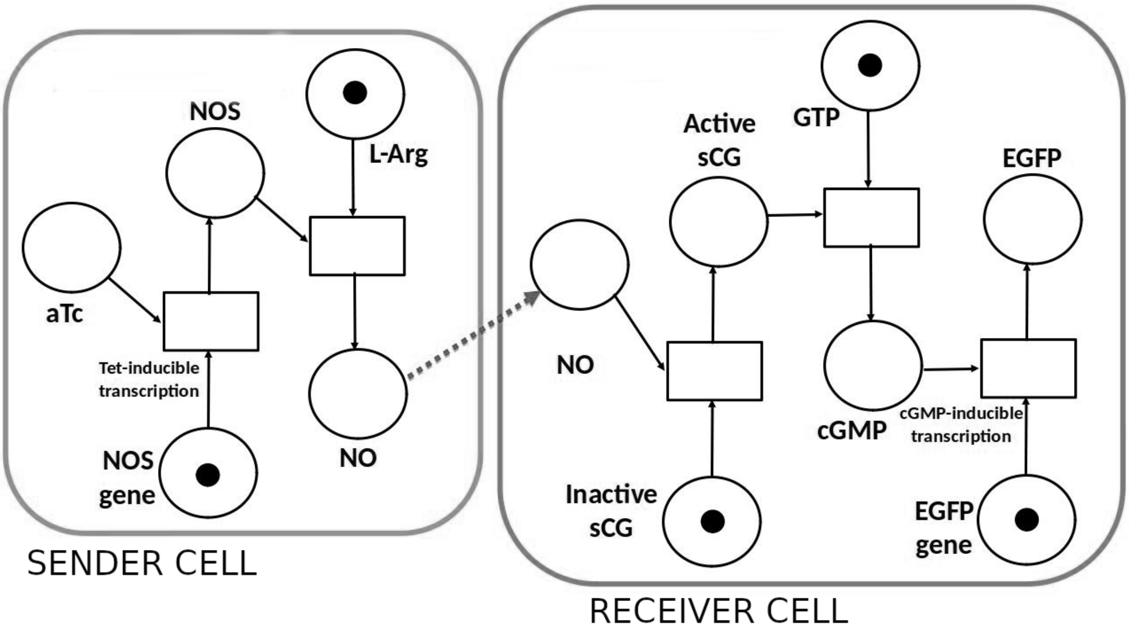


Figure 4.31. The NWN model at the bottom level represents the cell types involved in the system. They are described using Petri Nets modeling the relevant synthetic functionalities that they implement. Sender cells model the aTc-dependent transcriptional activation of NO synthase with the enabling rule of the transition modeling the production of the NOS gene product. The marking of the NOS operates the transition modeling NO production from L-Arg. Receiver cells depend on the marking of the place for incoming NO molecules. These activate the sCG enzyme, which, if it is present in its active form, corresponds to a marking enabling the transition transforming GTP into cGMP. This transition activates through a dedicated enabling rule the transition for the transcription of the EGFP gene. Adapted from Bardini, Politano, et al., 2018

Heterogeneous cell culture as a functional module

The model easily covers the fundamental mechanisms characterizing this partly artificial system employing the basic building blocks from the modeling approach proposed for ontogenetic processes.

The ISG models different positions in the cell culture medium, and mediates the administration of signals such as aTc to the cells. Positions may host either cells or culture medium.

aTc signaling relies on a form of hierarchical signaling, which the DL model operates.

NO signaling relies on *molecular flow* for modeling the diffusion of the NO molecule in the culture medium.

The DL model covers the states the receiver cell can be in; that is, it can either produce the fluorescent signal or not.

aTc administration draws a particular kind of trajectory for the Cell models. In fact, instead of evaluating conditions from them, the *checkpoint* in this case imposes a state change according to conditions from its level. This capability for imposing conditions over the system reflects the artificial part of the culturing process, over which the experimenter has effective control.

Fluorescent reporting is modeled similarly to a *differentiative trajectory*: the receiver cells can pass to the ”fluorescent” state after a *checkpoint* assesses their production of EGFP in the dedicate place from the receiver place.

The Cell models, for the sender and the receiver cells respectively, represent *cell autonomous* mechanisms, except for they communicate with the ISG and the DL. In this case study, the Cell models cover the artificial constructs to embed in the cells only.

NO production in the sender cell relies on a *signal sensing* building block for receiveing the aTc signal from the ISG model. Then, an *activating gene regulation*, *transcription* and *translation* blocks operate the production of NOS from its gene. An *enzymatic reaction* block models the NOS-mediated production of NO from L-Arg. A *signal sending* block takes care of sending the NO signal back to the ISG.

EGFP production in the receiver cell relies on a *signal sensing* building block for receiveing the NO signal from the ISG model. Then, NO enables an *enzymatic reaction* activating sCG, which, with a second *enzymatic reaction* block, produces cGMP from GTP. GTP, in turn, enables a *transcription* and a *translation* building block for the EGFP gene.

4.6 A modeling approach to epidemiology

To prove the semantic versatility of the presented modeling approach, this Section illustrates an entirely different application example, where models deal with bacterial cells, microbiotas and their hosts.

4.6.1 Application example: the spread of antibiotic resistance within the microbiota

Poor management of antibiotics administration in clinical practice and intensive breeding seems to have caused, in the last years, the emergence and increase of resistance in several bacterial strains. Policies and practices for antibiotics managements in different areas of the societal system point to a multi-level problem: considering each as a meta-organism, it is necessary to consider the microbiota living with it, which plays a role in the spread of antibiotic resistance. Each microbiota, includes bacteria, fungi, viruses, and other microbial and eukaryotic species. Their population growth dynamics, together with exchanges of genetic information among microorganisms can both facilitate resistance spreading and open possibilities for its prevention.

A possible intervention point is the design of more efficient antibiotic administration protocols, preventing or slowing down the spread of antibiotic resistance, both within single microbiota and at the hosts' population level.

The microbial taxa associated with humans exhibit great genomic diversity Ley et al., 2006, corresponding to excellent enzymatic capability and can control the physiology of the host. The host organism and the microbiota live in a symbiotic relationship. Bacteria get to live in a favorable environment for them, and the host organism gains a diversity of advantages M. Li et al., 2008. For example, the microbiota often can protect the host from pathogens Kamada et al., 2013.

Each microbiota has different proportions of different bacterial species, making overall growth dynamics emerge. The overall availability of resources bounds the overall growth and the way they occupy the available ecological space Korem et al., 2015. The larger a bacterial population in the microbiota, the more its genetic profile determines the functional profile of the overall population. Healthy microbiotas have a necessary core of functionalities, and at the same time, they show impressive species diversity, which makes them capable of adaptation and plasticity Turnbaugh et al., 2007.

It is possible that brand new functional capabilities emerge in a microbiota, thanks to different mechanisms. For example, previously absent species join in from the environment, starting to contribute to the pre-existing microbiota Lozupone et al., 2012. Another way is that thanks to the activation of *Horizontal Gene Transfer* (HGT) mechanisms, the acquisition of new functional capabilities can take place across species within the microbiota Thomas and Nielsen, 2005, which usually happens via plasmids exchange between cells Huddleston, 2014.

Antibiotics can cause insurgency and spread of resistance in all the bacterial populations they come in contact with after administration. They target conserved biological features Arenz and D. N. Wilson, 2016 in bacterial cells, and then they hit a broad spectrum of different bacterial species at the same time. Antibiotic administration kills both pathogenic and non-pathogenic species in the microbiota. This lack of species-specificity selects for survival only resistant species, that after the death of non-resistant cells can access the large portion of the ecological space which is freed from them, having less competition for resources Rodríguez-Rojas et al., 2013.

HGT promotes diversity in the bacterial population and improves their overall resilience by the sharing of capabilities. The HGT-mediated acquisition of antibiotic resistance is, for the cells involved, a form of adaptation to an environment becoming hostile for the antibiotic treatment.

In this application example, the proposed modeling approach targets a different multi-level system than the previous examples. The modeling objective, in this case, is to both better understand the spread dynamics, and to find antibiotic administration strategies allowing to slow down or prevent the spread of resistance within a microbiota and the hosts' population Lozupone et al., 2012.

Model organization

Compared to the previous application examples, in this case, the role of state representation is prominent respect to that of spatiality. Also, there is no detailed representation of cell-autonomous mechanisms: colored tokens model cells, providing information about their species and resistance state (see 4.32). As depiced in Figure 5.1, the model represents three system levels:

1. the population of hosts;

2. the microbiota;
3. the bacterial cells;

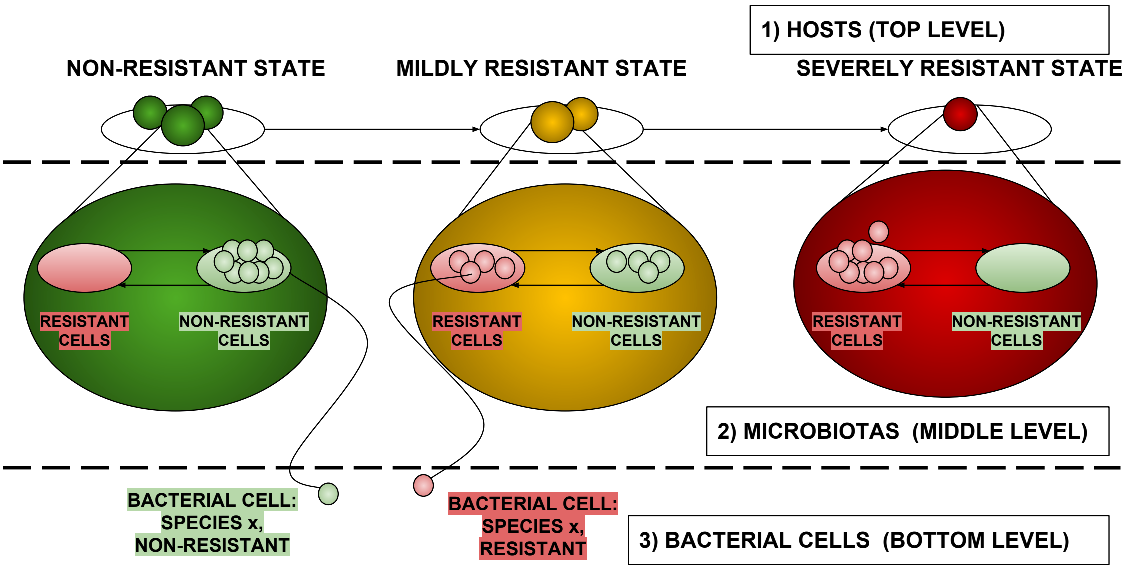


Figure 4.32. High level conceptual view of the proposed computational model organized into three levels. Taken from Bardini, Di Carlo, et al., 2018.

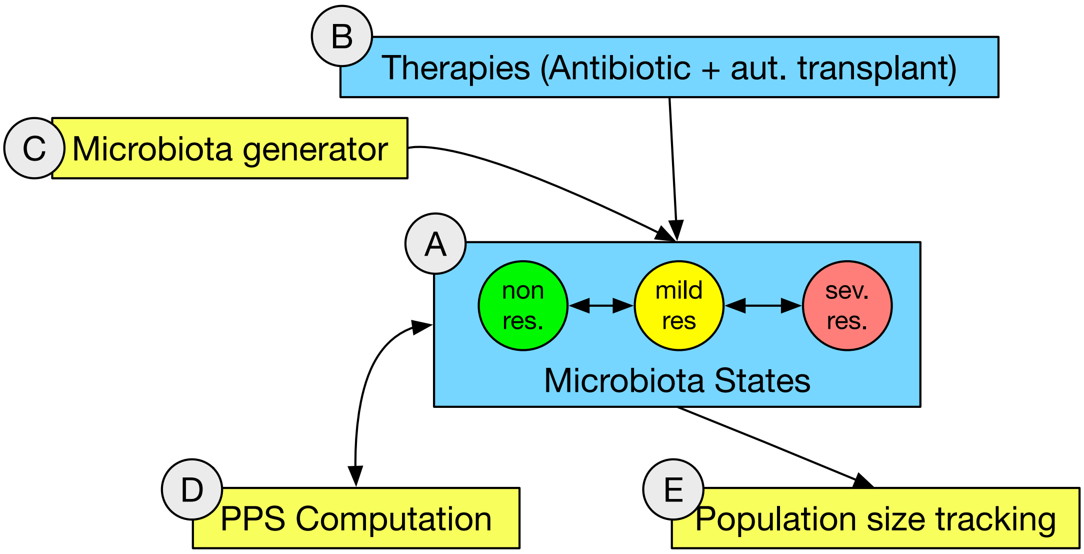


Figure 4.33. Conceptual view of the network architecture for the hosts (top) level. Three main places describe the health states the microbiotas can assume. The letters identify the different sections of the detailed model presented in 4.34. Taken from Bardini, Di Carlo, et al., 2018.

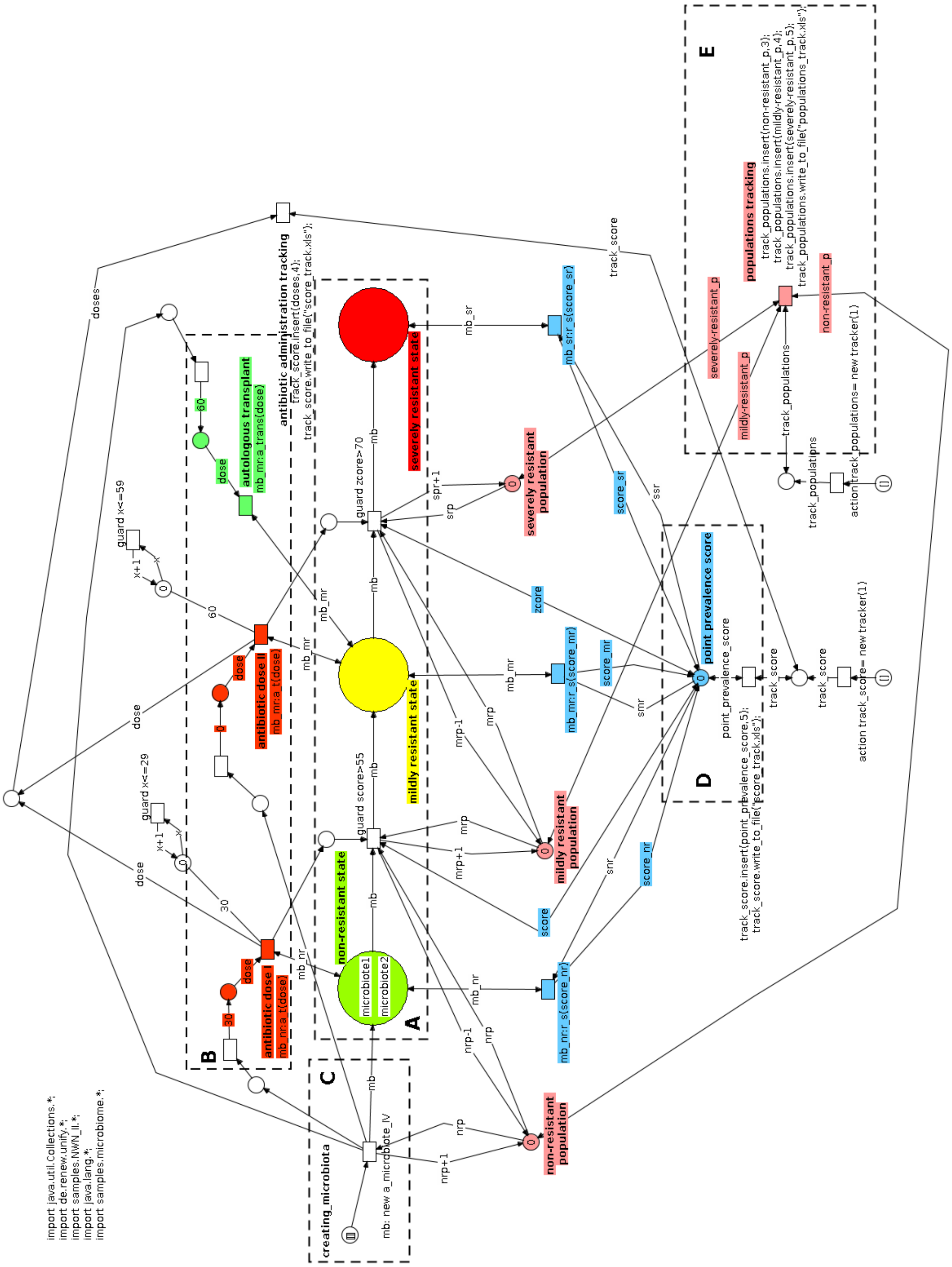


Figure 4.34. Network architecture for the top-level States Landscape model, modeling populations of hosts. Three main places describe the health states the microbiotas can assume: the non-resistant (in green), mildly resistant (in yellow) and severely resistant (in red) states respectively (A). Transitions can move microbiota tokens, each having the network structure from 4.36, to the next place. The state of non-resistance holds two microbiota tokens depicted in a compact form, i.e., with their name only. According to the value of their point prevalence score which a synchronous channel (D) reads from networks at the lower level, possibly taking the relative microbiota to the next step along with resistance progression (the structures in E track the changing numerosity of microbiota instances in each place). Synchronous channels take care of the antibiotic administration and microbiota integration events (B), activating network structures at the lower level (4.36.B and 4.36.F, respectively) according to time delays and the number of microbiota instances injected in the network by the dedicated structure (C). Taken from Bardini, Di Carlo, et al., 2018.

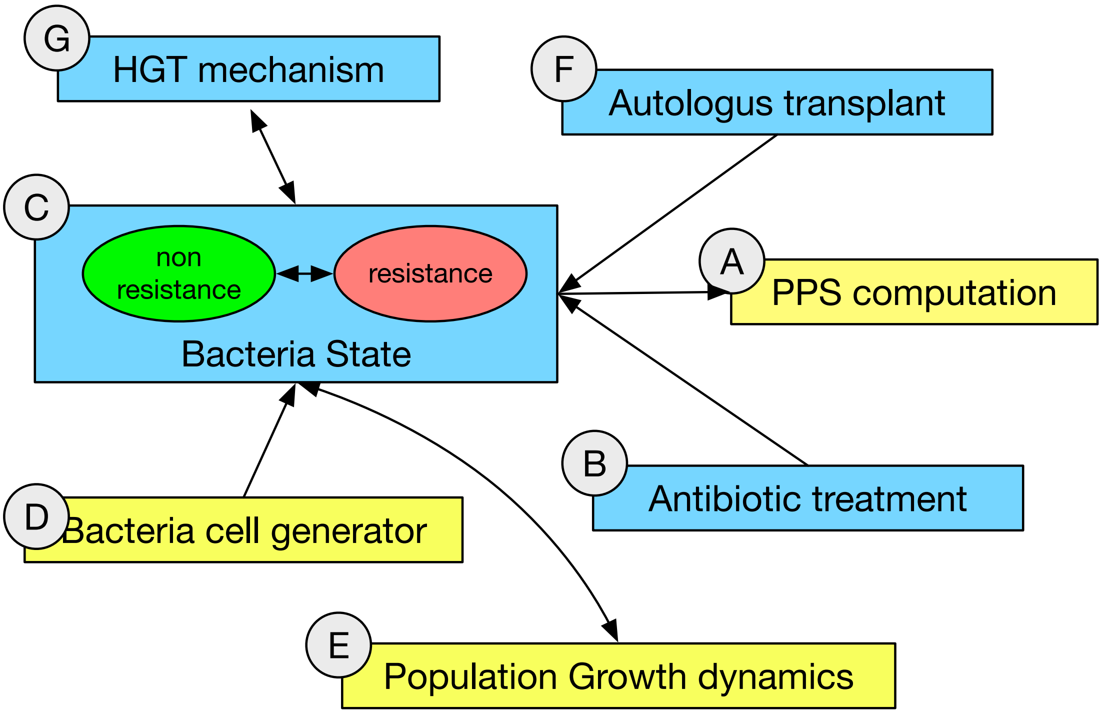


Figure 4.35. Conceptual view of the network architecture for the microbiotas level: two main places describe two conditions each bacterial cell can assume. The letters identify the different sections of the detailed model presented in 4.36. Taken from Bardini, Di Carlo, et al., 2018.

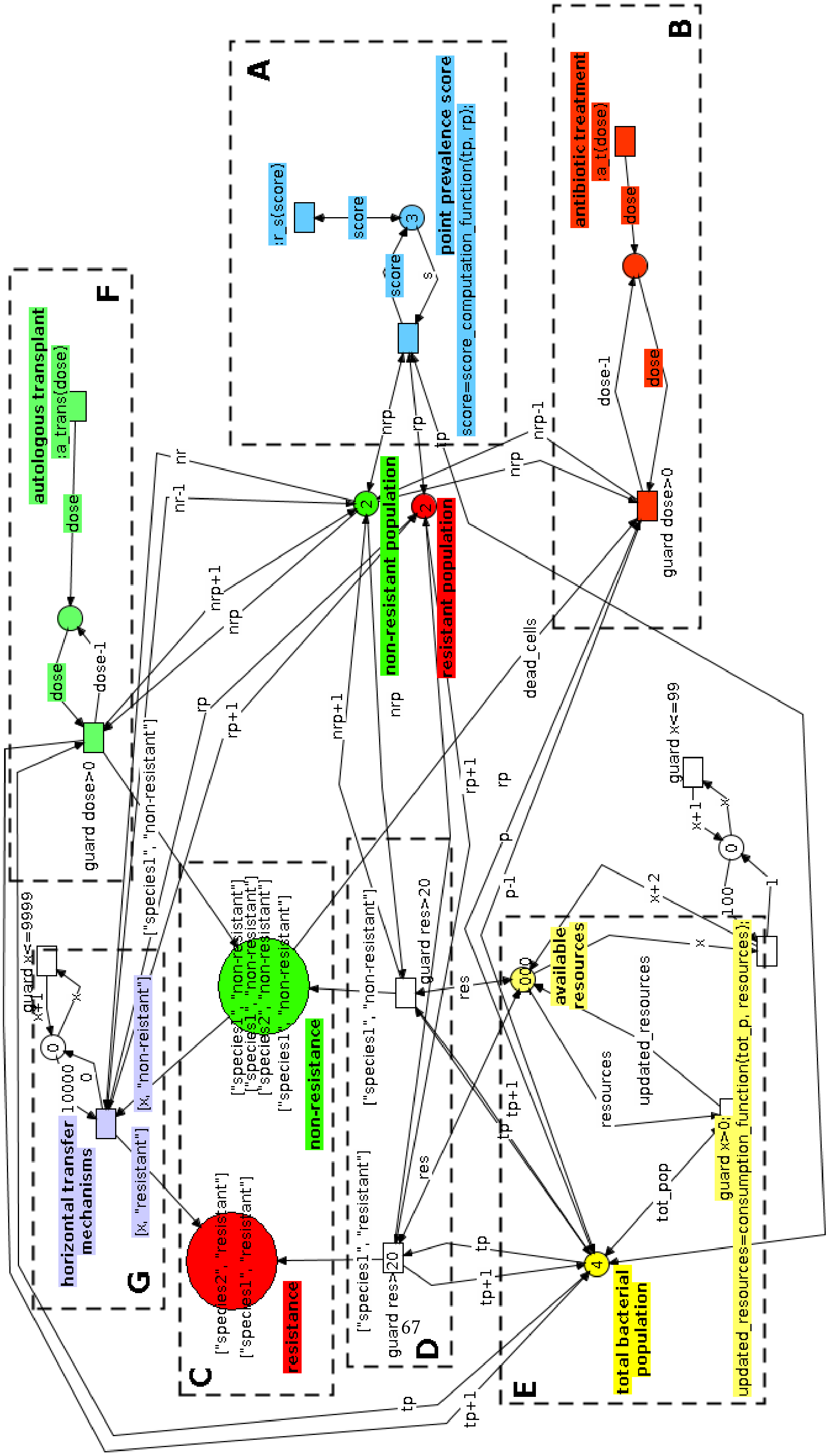


Figure 4.36. Network architecture for the bottom-level States Landscape model. It models individual microbiota. Two main places describe two conditions each bacterial cell can assume: a state of non-resistance (green) and a state of resistance (red), respectively (C). Horizontal transfer mechanisms can turn non-resistant cells into resistant ones (G). Network structures managing the generation of new bacterial cells (D), total population

Genetic exchanges in the microbiota as a functional module

The building blocks from the proposed modeling strategy in this context adapt to the actors involved in antibiotic resistance spread and their main functional interactions Bardini, Politano, et al., 2017b; Centers for Disease Control and Prevention, 2013.

The ISG reduces to a minimum: no spatial resolution is provided for the simulation of interactions between bacterial cells. The DL model is, like in the previous example, better interpreted as a Landscape of States. There are two States Landscape models. One for the microbiota, and another one for the bacterial cells. At the top level, the emergent behavior of its bacterial population determines the state of each microbiota.

Increases in resistance level for the host leverage a building block similar to *differentiative trajectories*: the host organism moves from a place corresponding to a lower resistance level to one for higher resistance. A *checkpoint* building block regulates the passage, reading from the lower level a dynamically computed metric evaluating the overall population state. That is, the Point Prevalence Score (PPS), i. e. the proportion of bacterial cells carrying resistance factors over the total bacterial population Szklo and Nieto, 2014 (see Figure 5.7.A).

Treatment administrations leverage a form of hierarchical signaling operated by the States Landscape model, determining the marking of lower-level nets with a precise temporal organization (see Figure 5.6.B) in all microbiota instances existing in the target place. Antibiotic doses deplete non-resistant cells, while reintegration waves make new non-resistant cells join the microbiota. Both events dramatically affect population dynamics.

At the middle level, bacterial cells can live either in resistant or non-resistant states. Also, they can carry graded levels of resistance as colored tokens.

HGT mechanisms leverage a building block similar to *differentiative trajectories*: the bacterial cells move from a place corresponding to the non-resistant state to one for the resistant state. The passage is regulated by an active *checkpoint* building block, which can transform non-resistant cells in resistant ones, moving them in a dedicated place. This block enables after a stochastic time delay, modeling the probability for resistance acquisition for the average bacterial cell in the microbiota.

Bacterial duplication leverages a mechanism similar to the *Mitosis* building block. Depending on the availability of ecological space within the microbiota, which growth rates of subpopulation and overall availability of resources allow to compute, bacterial cells can divide.

Bacterial death leverages a mechanism similar to the *Apoptosis* building block. Depending on the entity of antibiotic dose administered, which arrives from the top States Landscape model through a *signal sensing* building block, a number of non-resistant bacterial cells die.

The Cell models are, in this case, colored tokens which can carry information about the bacterial species and the state of resistance of the single cell.

General experimental designs

To provide an example of the types of experiments that a model specified under this approach can support, in this section three different experimental conditions initialize simulations. Each of them corresponds to a different antibiotic administration procedure. The objective is to compare the effects of different administration protocols over the spread of resistance. On one side, the simulation of traditional protocols takes place. On the other side, existing clinical practices such as autologous microbiota transplants also called bacterial therapy Lozupone et al., 2012. This strategy has the scope to mitigate the advantage acquired by the resistant cells in colonizing the gut niche, reinforcing the non-resistant population.

- no antibiotic administration is performed (Figure 4.37);
- two administration events take place, the first one devising a lower dose and the secondo an higher one (4.38);
- the application of an innovative treatment protocol, where administration of the first low antibiotic dose pairs with reintegration of non-resistant bacterial cells in the microbiota (4.39).

In this first experimental design, simulations track the PPS of a single microbiota; this measures the prevalence of resistant cells within the total bacterial population. PPS score is sampled at different stages of the simulation:

- before any treatment;
- after the administration of the first, low dose of antibiotic (plus, for the innovative protocol, the microbiota integration)
- after the administration of the second higher dose of antibiotics.

The values presented in Figures 4.37, 4.38, 4.39 are computed as the average point prevalence score (APPS) over 30 simulation runs. In this way, the proposed model intends to enable the study of the overall resistance state of microbiota and the dynamics of its insurgency and the different treatment protocols.

In the first experimental condition, devising no treatment at all, the APPS remains at a low and almost constant level during the whole simulation. Small increases can occur after the activation of random mutations or horizontal gene transfer events. In only 4 out of 30 experiments the score exceeded the threshold required to assign the microbiota into a state of mild resistance. The relative averaged APPS of 57.75±0.63 (second bar of 4.37) refers to these four cases. All remaining simulations find representations in the first bar of 4.37, showing that the microbiota remained in a healthy state, with an APPS score of 48.67±0.61. In none of the simulations, the microbiota reached a state of severe resistance.

In the second experimental condition, the model simulates a treatment protocol composed of two administrations of an increasing dose of antibiotic (4.38). Before treatment (as in the control condition), the APPS was 50.34±0.88. The administration of the first lower dose allows partial recovery of the non-resistant portion of the bacterial population (APPS of 64.74±1.14); the second higher dose, instead, takes the microbiota towards a state of increased resistance (APPS 93.24±2.4).

In the third experimental condition (4.39), the previous treatment protocol combines with a parallel preventive reintegration of non resistant bacterial cells; this improved protocol has a significant effect after each antibiotic dose: the APPS decreases of 18.08% after the first dose (when compared to that originating from the traditional treatment alone), and decreases of 24.12% w.r.t. the traditional protocol after the second antibiotic dose.

PPS remains constant when no treatment is administered (4.37). With traditional treatment, it increases in correspondence with the administration of the two antibiotic doses (blue curve in 4.38). The same dynamic emerges in 4.39, with the difference that the preventive action of bacterial reintegration mitigates the resistance effects of the first dose of antibiotics, taking the PPS score to a level similar to pre-treatment.

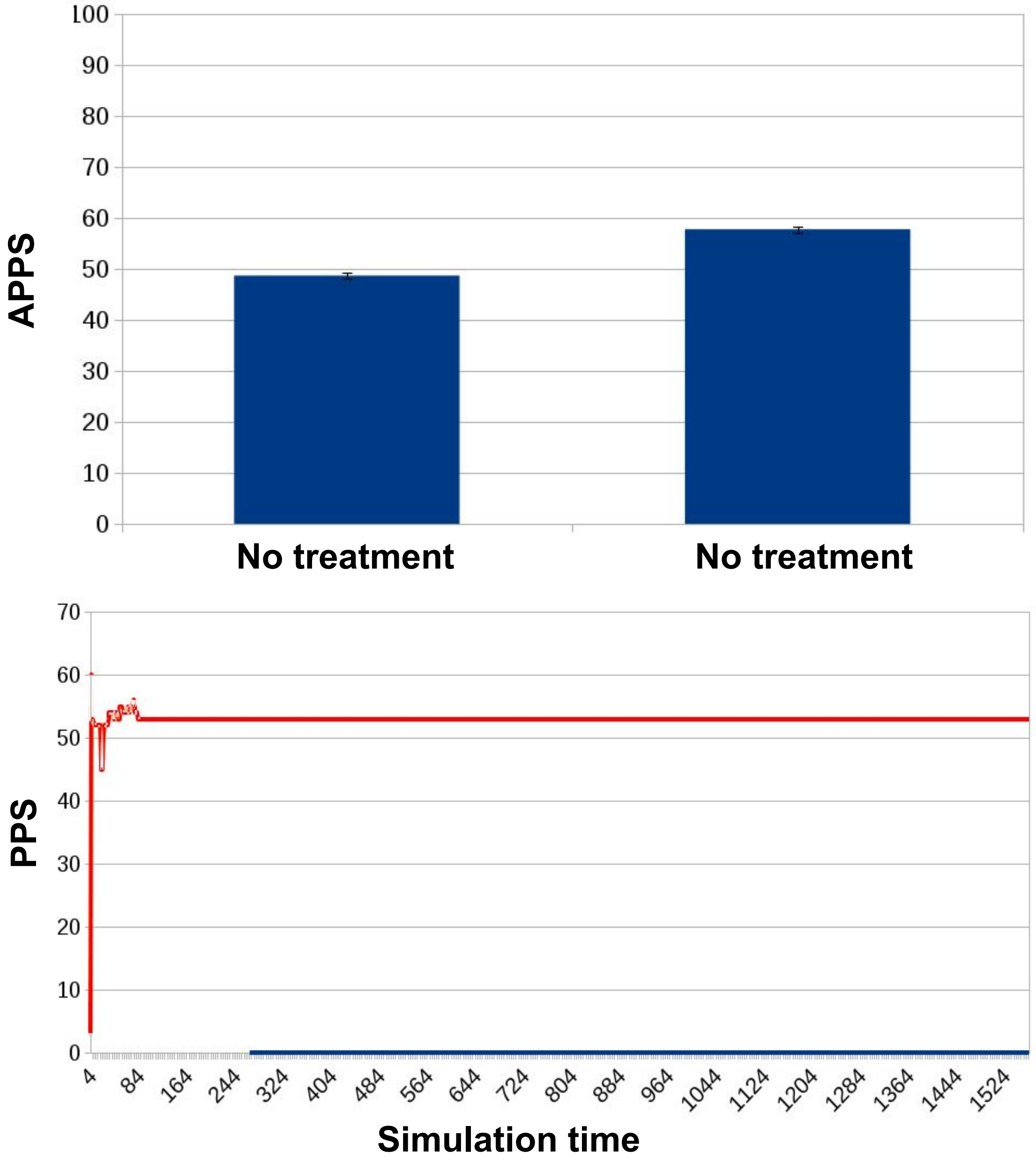


Figure 4.37. Results for ED1 in case of no treatment. The APPS (Average PPS) is 48.67±0.61 (first bar), under the threshold of mild resistance, keeping the microbiota into the ”non-resistant” state. Only in 4 cases such threshold was crossed, and the APPS for those cases is slightly higher: 57.75±0.63 (second bar). All cases had similar tracks, where PPS (the curve in orange) reaches a value and keeps it steadily along the simulation. The curve in blue tracks treatment administrations: it marks zero during the entire track. Taken from Bardini, Di Carlo, et al., 2018.

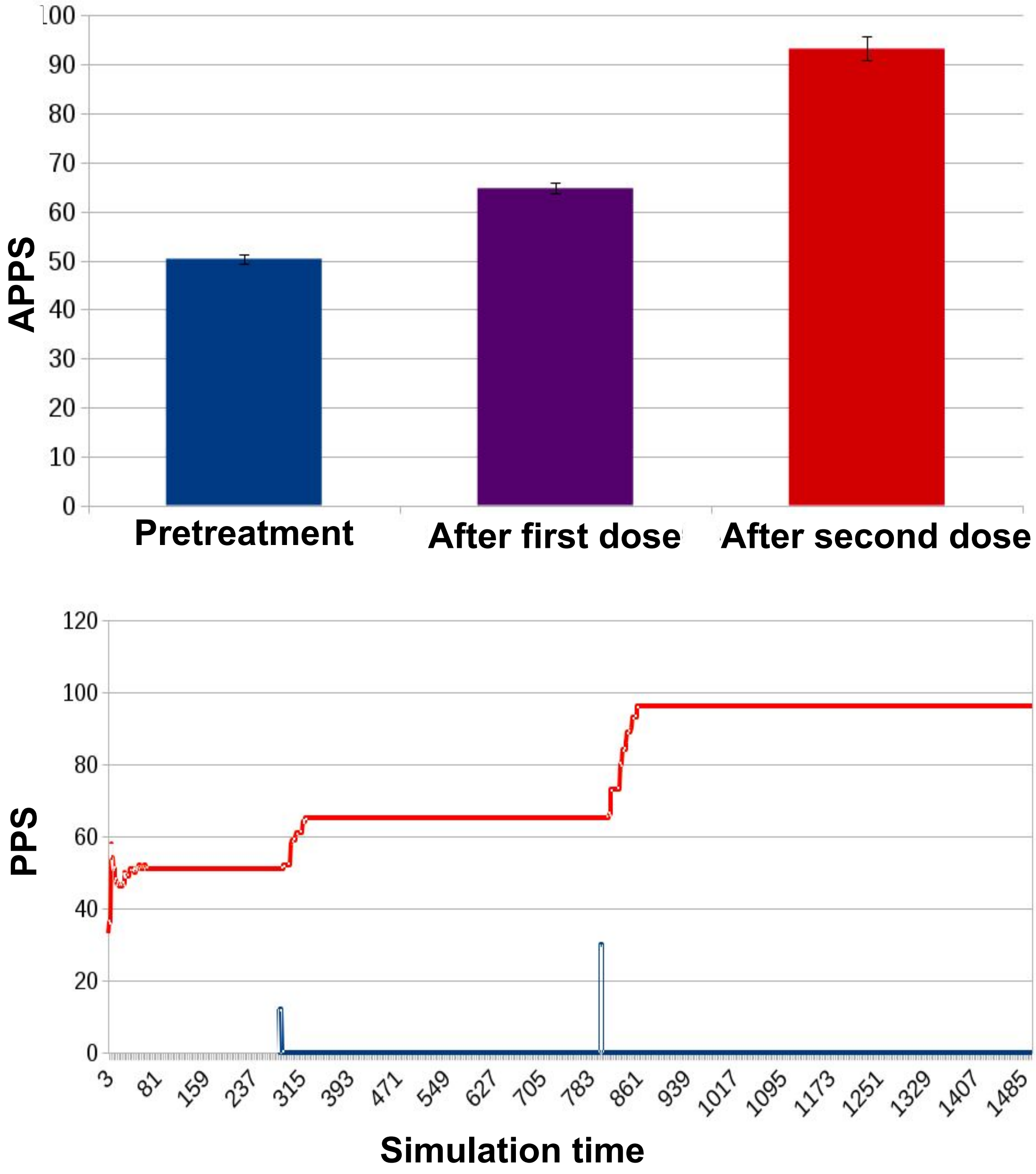


Figure 4.38. Results for ED1 in the case of traditional treatment. Two doses of antibiotics target the microbiota, the first lower and the second higher. The APPSs for the three important stages of the experiment are 50.34±0.88 before treatment (homologous to that in 4.37), 64.74±1.14 after the first dose (beyond the threshold for mild resistance) and 93.24±2.4 after the higher dose, beyond the threshold for severe resistance. The simulation track of the PPS shows that after each dose (the curve in blue, whose peaks represent doses administration) PPS increases proportionally, moving from the ”non-resistant” steady state to the ”mildly resistant” one, and finally to the ”severely resistant” state, where it stays. Taken from Bardini, Di Carlo, et al., 2018.

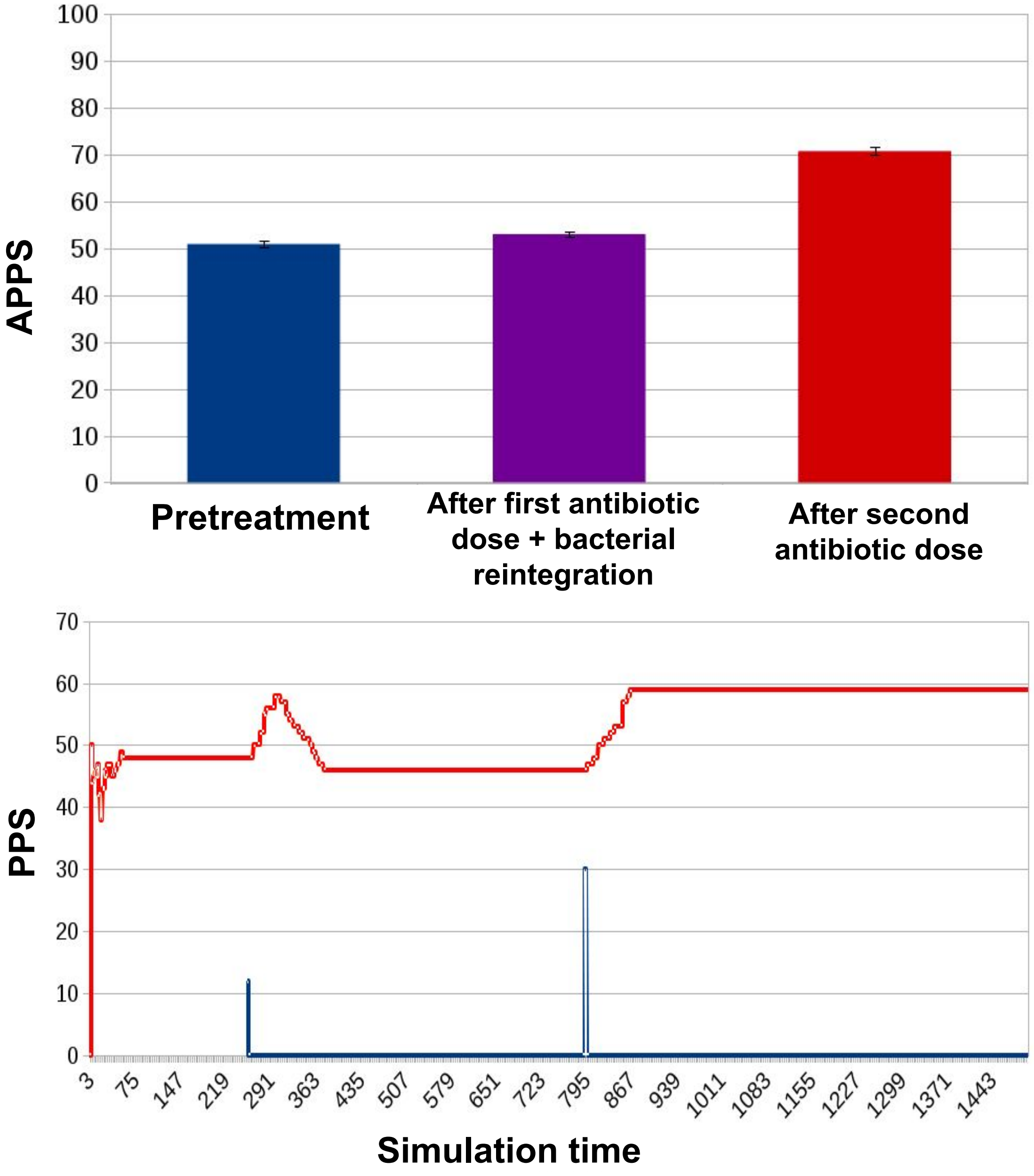


Figure 4.39. Results for ED1 in case of the innovative protocol. This protocol lowers APPS both in the second and in the third stages of the experiment: while pretreatment APPS remains homologous to those in 4.37 and 4.38 (50.94±0.67), after the first dose of antibiotic and microbiota reintegration it drops to 53.03±0.52. Eventually, after the second dose of antibiotics, it reaches 70.75±0.82, corresponding to significant decreases compared to the corresponding simulation stages in the traditional treatment scenario 4.38. In the simulation curve, we observe how the microbiota reintegration counterbalances the effects of the first antibiotic dose on PPS: in the first place PPS begins to rise, but it is bounded right away to a low level by the preventive action, leaving on the track just a transient spike. After the second, higher dosage of antibiotics, PPS increases, reaching a steady state at a higher level, which is lower than that reached in 4.38. Taken from Bardini, Di Carlo, et al., 2018.

A second experimental design centers over the host population level: multiple instances of the microbiota model a population of hosts. The following experimental conditions were taken into account:

- no treatment administration;
- a single, high-dosage treatment administration (see Figure 4.43);
- a single, high-dosage antibiotic administration, combined with bacterial reintegration (see Figure 4.40).

We simulated 50 instances of the microbiota network, collecting the simulation times at which the number of severely resistant hosts exceeds the number of mildly resistant hosts (severe resistance onset time, T), representing the fact the spread reached a turning point, after which extensive resistance diffusion takes place.

When no treatment administration, severe resistance does not emerge. Figure 4.43 shows how a high-dosage antibiotic administration determines an average onset time (AT) of 162.87±7.78, while bacterial reintegration, AT is delayed on average by 9.68% in AT, 178.64±7.4 (see Figure 4.40).

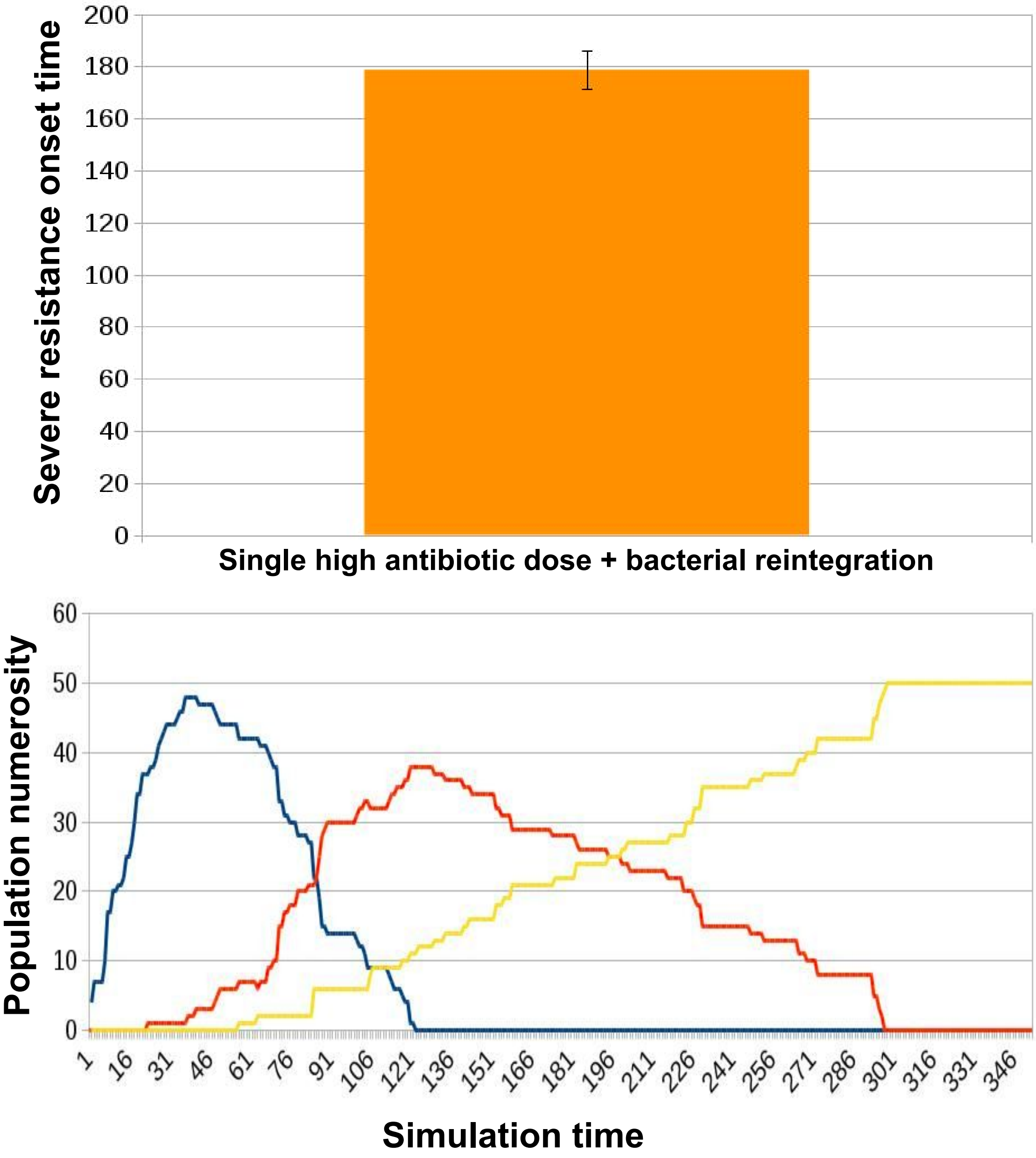


Figure 4.40. Results for ED2 in case of innovative treatment. A single high dose of antibiotic combined with bacterial reintegration reaches the microbiota. Microbiota reintegration combined with antibiotic administration yields an increased AT (178.64±7.4), compared to that observed with the administration of the antibiotics alone (Figure 4.43). In simulation tracks we notice how the overall migration of healthy individuals towards a worsening resistant state slows down, resulting in a slower severe resistance onset time (T). The blue curve represents the number of non-resistant individuals, the orange curve the number of mildly resistant individuals, and the yellow one the population of severely resistant individuals. Taken from Bardini, Di Carlo, et al., 2018.

The role of bacterial predation in the spread of resistance

This section presents an application example of the general modeling approach to resistance spread dynamics. This general framework applies to the spread of antibiotic resistance in the mouse gut microbiota. This model also considers the role of bacterial predation mechanisms Acinetobacterium and E. Coli bacterial cells engage in Asahara et al., 2016; Cooper RM, 2017.

Among other bacterial species that can live in the microbiota, Acinetobacterium baumanii acts as an opportunistic pathogen, posing a real threat to immunocompromised or injured individuals. This species thrives in hostile environments, such as hospitals or battlefields, because it can survive in biotic and abiotic environments thanks to its capability to build up a protective biofilm. For this reason, cells survive in the environment for a longer time. This improved survival capabilities support the formation of pathogenic reservoirs, making infection by environmental contact more probable. In addition, Acinetobacterium Baumanii acquires antibiotic resistance very fast (Antunes et al., 2014; Joly-Guillou, 2005), reaching multi drug-resistance (MDR) rates beyond 60% (Antunes et al., 2014; Centers for Disease Control and Prevention, 2013).

A model integration procedure embeds quantitative parameters (e.g., bacterial population sizes and HGT rates) from the literature and plugs them into the proposed model. For this reason, this applicative case also functions as a valuable example of the capability of the proposed formalism to integrate information of different types from different sources.

In particular, information flows into the microbiota model and the hosts' population model from two experimental works treating the problem from the two different aspects respectively Cooper RM, 2017 Asahara et al., 2016. In Cooper RM, 2017, authors claim what makes Acinetobacterium faster in acquiring resistance is a particular HGT mechanism: *bacterial predation*, which involves a predator species that kills adjacent prey cells and acquires their genes, including those coding for adaptive resistance. In Asahara et al., 2016, the authors study the same bacterial species, but they focus on the spread and infectious activity of MDR Acinetobacterium strains in the murine gut microbiota. They aim to provide a *in vivo* model of post-surgery infections in hospitals. Also, they assess how a bacterial reintegration-based therapeutical approach can buffer and mitigate the rise of MDR Acinetobacterium, mitigating the infection and improving the general conditions of the host.

The presented model aims to investigate how Acinetobacterium, through bacterial predation, contributes to the overall resistance state of the microbiota and the prevalence of Multi-drug Resistance (MDR).

The model represents three organization levels from the system:

- a group of mice hosting the microbiota;
- their microbiota;
- the different bacterial cells from the microbiota.

At the top level, treatment administration (specified by the experimental design) takes place for murine hosts. In general, possible treatments devise antibiotic doses, possibly combined with bacterial reintegration.

The microbiota model aims to represent, in a very simplistic way, the diversity inherent to the population of microorganisms involved. See Appendix 3 for further details on the net architecture and implementation. Figure reffig:microbiota2-conceptual shows its high-level structure while Figure 4.42 reports its detailed implementation.

At this level, Acinetobacterium cells acquire MDR making use of both standard HGT mechanisms (4.42.G) and bacterial predation (4.42.B) over other bacterial species, which can occur at variable rates Cooper RM, 2017. The model explicitly covers both mechanisms. Dynamically computed scores represent the resistance state of the microbiota: the Point Prevalence Score (PPS, see Section 4.6.1), and the Acinetobacterium Resistance Level (ARL, the average amount of antibiotic-resistant DNA factors acquired through bacterial predation divided by the number of total predation events), representing the probability of a population to acquire multi-drug resistance factors. In order to properly integrate kinetic parameters from Cooper RM, 2017 into the model, the model provides an explicit representation of time.

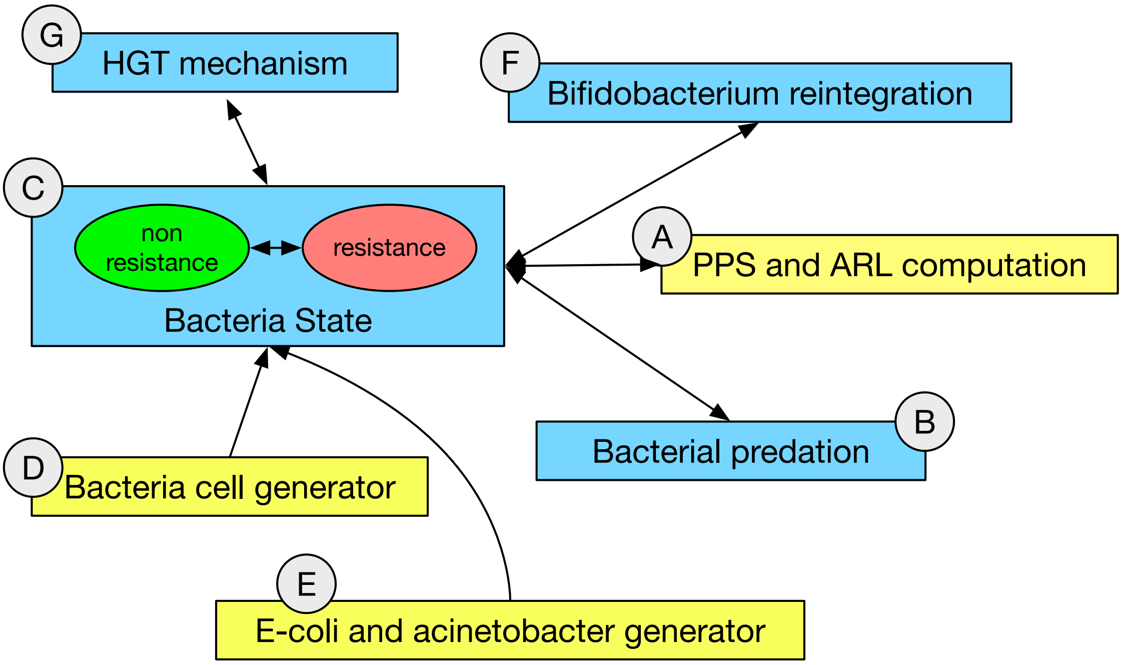


Figure 4.41. Conceptual view of the network architecture for the microbiotas (median) level in the Acinetobacterium/E.coli model. Taken from Bardini, Di Carlo, et al., 2018.

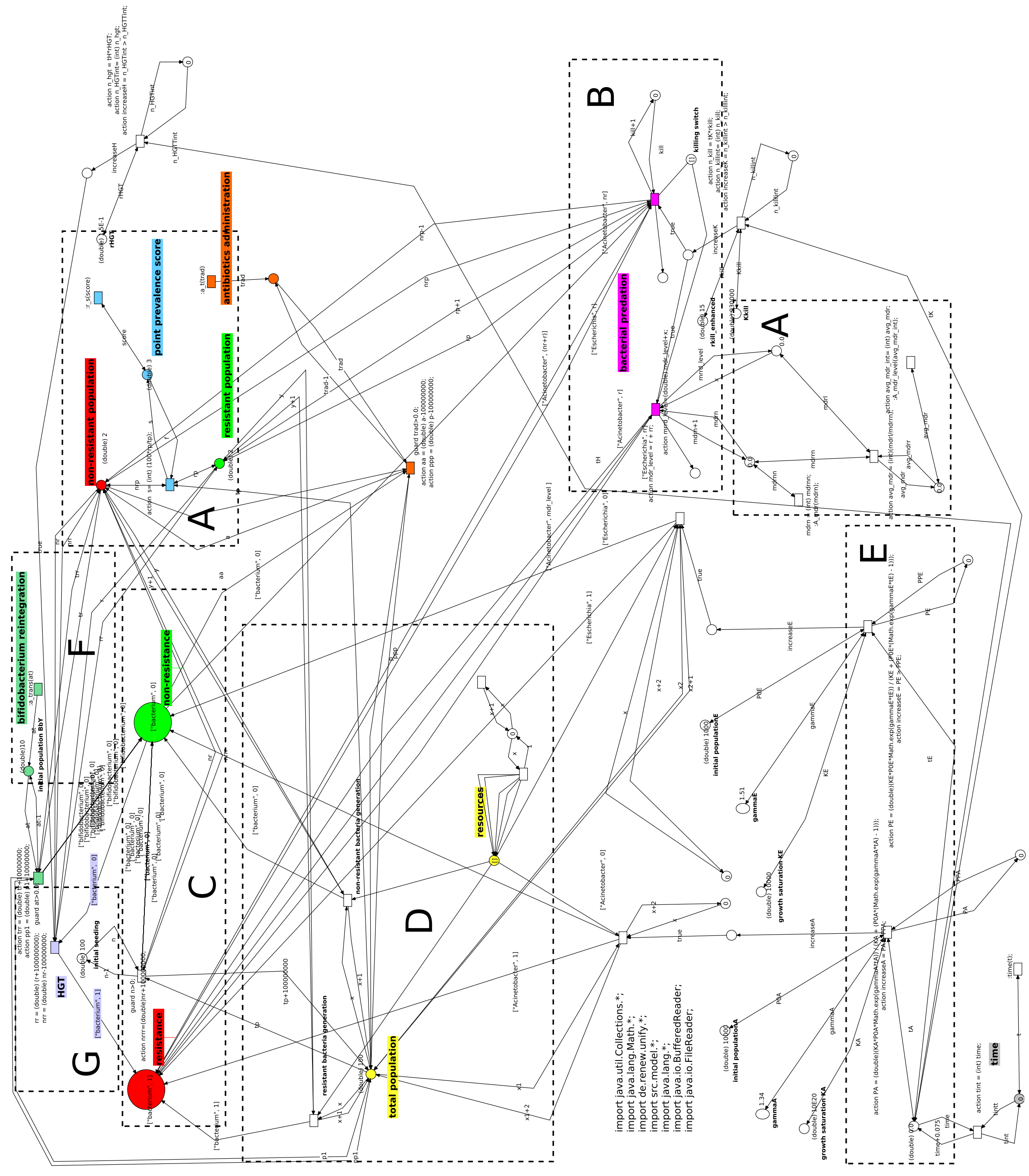


Figure 4.42. Network architecture for the microbiotas (median) level in the Acinetobacterium/E. Coli model. Two main places describe two conditions each bacterial cell can assume: a state of non-resistance (green) and a state of resistance (red), respectively (C). Horizontal transfer mechanisms can turn non-resistant cells into resistant ones (G). Network structures managing the generation of new bacterial cells, including Acinetobacterium and E. Coli (E), total population numerosity and resource availability (D) give rise to a competitive population dynamics between resistant and non-resistant populations. Dose-dependent depletion of non-resistant cells following antibiotic administration (B) and Bifidobacterium reintegration (F) events activate as synchronous channels with structures in 4.34.B. The structure in A dynamically computes the point prevalence score and the Acinetobacterium Resistance Level of the microbiota network, making the information available for the upper level through a synchronous channel (see 4.34.D). Taken from Bardini, Di Carlo, et al., 2018.

Colored tokens which live at the microbiota level represent bacterial cells. The color, in this model, identifies the species. The species modeled explicitly are Acinetobacterium Baumanii, Escherichia Coli, and Bifidobacterium, or generic commensal bacteria. Color also represents the antibiotic resistance level of each cell, and only tokens which are modeling Acinetobacterium can cumulate resistance factors through bacterial predation.

Genetic exchanges in the microbiota: a specific application

The functional module presented in 4.6.1 finds further specification by modeling the role of bacterial predation in the spread of antibiotic resistance within the murine microbiota. The building blocks adapt to the specificities of this application example, embedding information and data from the selected sources.

This model as well is mainly oriented to representing states rather than spatial aspects from the system. There are two States Landscape models: one holds the murine microbiota, and the other one the bacterial cells. At the level of murine hosts, organisms can be in different resistance states, also according to the treatments they received.

Increases in resistance level for the host leverage a building block similar to *differentiative trajectories*: the murine hosts move from lower to higher resistance states by the activity of a *checkpoint* building block, reading from the lower level the state of the microbiota, which computes it dynamically as the PPS score.

Treatment administrations leverage a form of hierarchical signaling operated by the States Landscape model. It manages antibiotic doses, possibly in combination with microbial reintegration, with a precise temporal organization.

In each microbiota, bacterial cells can live either in resistant or non-resistant states. Also, they can carry graded levels of resistance.

Standard HGT makes any bacterial cell move from a place corresponding to the non-resistant state to one for the resistant state. The passage is regulated by an active *checkpoint* building block, which can transform non-resistant cells in resistant ones, moving them in a dedicated place. This block enables after a stochastic time delay, modeling the probability for resistance acquisition for the average bacterial cell in the microbiota.

Bacterial predation makes an Acinetobacterium cell consume any other cell, and, if present, acquire its resistance factors, possibly cumulating it with other ones it had already. The event is regulated by an active *checkpoint*, which enables after a time delay computed by kinetic parameters from Cooper RM, 2017. Also, it requires the presence of a predator and a prey cell, respectively. A bacterial cell, either resistant or not, is subtracted from the microbiota, while an Acinetobacterium cell possibly acquires a first, or additional resistance factor.

Bacterial duplication leverages a mechanism similar to the *Mitosis* building block. Depending on the availability of ecological space within the microbiota, which growth rates of subpopulation and overall availability of resources contribute to computing dynamically, bacterial cells can divide. For E. Coli and A. Baumanii cells, duplication rates correspond to kinetic parameters from Cooper RM, 2017.

Bacterial death leverages a mechanism similar to the *Apoptosis* building block. Depending on the entity of antibiotic dose administered, which arrives from the top States Landscape model through a *signal sensing* building block, a number of non-resistant bacterial cells die.

The Cell models are, in this case, colored tokens which can carry information about the bacterial species (E. Coli, A. Baumanii, Bifidobacterium, or generic commensal bacterium) and the state of resistance of the single cell. In the case of A. Baumanii, the resistance state is graded and can represent different levels of MDR.

Model integration

This applicative example illustrates the capability of the presented modeling approach for integrating information from different sources. The model embeds mathematical descriptions from Cooper RM, 2017. Table 4.2 recapitulates the main parameters employed. The initial conditions used in Asahara et al., 2016 are used to set up initial markings for simulations, to depict the untreated individuals. The model simulates a gram of microbiota. Optimal relative population densities between predator and prey can make killing enhancement reach a factor of 3 Cooper RM, 2017. For making this behavior emerge, in the model initial conditions are set so that the population sizes of Acinetobacterium and E.Coli recapitulate this proportion.

Table 4.2. Model parameters in the Acinetobacterium/E. Coli model. Quantities of bacterial cells are intended per gram of sample. Kinetic rates are intended per hour. Taken from Bardini, Di Carlo, et al., 2018.

Model parameters	**Values**	**Ref.**
P0tot	10e10	total bacterial population
P0ByB	10e2	Initial Bifidobacterium population
P0AB	10e2	Initial Acinetobacterium population
KA	10e20	Growth saturation for Acinetobacterium
KE	10e4	Growth saturation for E. Coli
gammaA	1.34	Growth rate for Acinetobacterium
gammaE	1.51	Growth saturation for E. Coli
killing rate (ehnanced)	150 h$\hat{-}$1	Frequency of predation events
HGT rate	0.15 h$\hat{-}$1	Frequency of Horizontal Gene Transfer events

4.6.2 Experimental design

Simulations compare the behaviors of an untreated microbiota, another one receiving antibiotic treatment, and a third one receiving antibiotic treatment combined with bacterial reintegration respectively, to study the spread of MDR in the Acinetobacterium cells and of resistance in the overall bacterial population.

The experiments follow a timeline recapitulating that from Asahara et al., 2016.

- at day 0 (in Asahara et al., 2016, day -7) daily administration begins, and it goes on until day 14.
- at day 7 (in Asahara et al., 2016, day 0, when infection with MDR Acinetobacterium takes place) relative bacterial population densities arise so to cause a killing enhancement of a factor of 2 in Acinetobacter, corresponding to increasing the probability that they acquire MDR.
- at day 14 the experiment is interrupted, and the variables of interest assessed.

The simulation of treatment includes antibiotic administration and, possibly, bacterial reintegration with Bifidobacterium 4.42.F.

Simulations cover 14 days. The model tracks both PPS and ARL scores at the end of the first, the seventh, and the fourteenth days in the simulation timeline.

In Figures 4.44, 4.45, 4.46 and 4.47, 4.48, 4.49 we separately present results for PPS and ARL tracking respectively.

Antibiotic administration causes average PPS to increase significantly, especially at day 14: while in the first experimental condition (no treatment), APPS is 72.29±0.07 (4.44), in the simulations including antibiotic treatment it reaches the value of 359.21±30.11 4.45. In 4.46 we observe how the combination of antibiotic administration and bacterial reintegration lowers the score both at day 7 and 14 compared to the case where antibiotics alone were administered; after seven days of the combined treatment APPS is 35.25±0.51, and at day 14 it reaches the value of 235.94±7.9.

As can be observed, average ARL (AARL) tends to increase along the simulation time, reflecting the growing pool of DNA exchanged through predation by Acinetobacterium. No significant variations are observed across experimental conditions (Figures 4.47, 4.48, 4.49), reporting that the MDR level in Acinetobacter grows steadily during the time of the experiments, as exemplified by the simulation tracks.

These results considered together are coherent with those presented in Asahara et al., 2016, where the effect of bacterial reintegration is measured evaluating the spread of resistant Acinetobacterium infection in the host during antibiotic treatment. They show reductions of the Acinetobacterium infection spread when bacterial reintegration combines with antibiotic treatment compared with the case antibiotic treatment is administered alone — considering that as an indirect measure for the resistance level in Acinetobacter, results indicate a 30% reduction of resistance level after bacterial reintegration. These results demonstrate, in this case, the capability of the presented model for making consistent predictions from and over the experimental data they integrate.

Even though antibiotic treatment and bacterial reintegration do not seem to affect the propensity of Acinetobacterium towards acquiring exogenous, potentially resistant DNA (as shown in Figures 4.47, 4.48, 4.49), they do affect the overall resistance level of the microbiota, and thus the probability of a predation event leading to the acquirement of multi-drug resistance takes place.

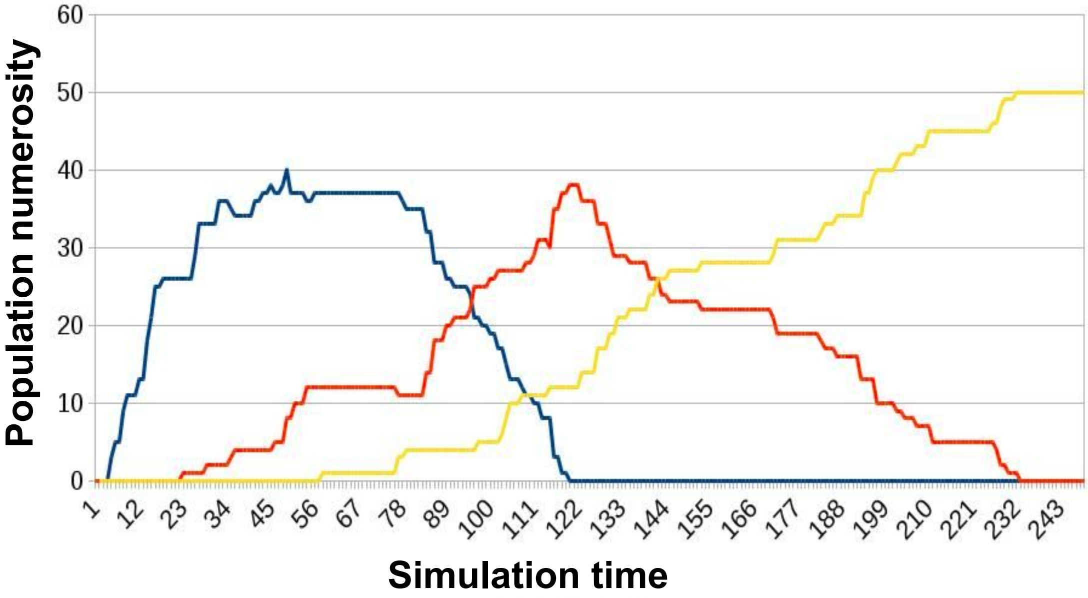


Figure 4.43. Results for ED2 in case of traditional treatment. A single high dose of antibiotic targets the microbiota. The administration of a single high dose of antibiotics reaches each microbiota instance, yielding an average onset time for severe resistance (AT) of 162.87±7.78. Simulation tracks from these experiments show healthy individuals (blue curve) progressively acquiring resistance. Some of them reach severe resistance right away (yellow curve), while most of them pass through a phase of mild resistance (orange curve). Taken from Bardini, Di Carlo, et al., 2018.

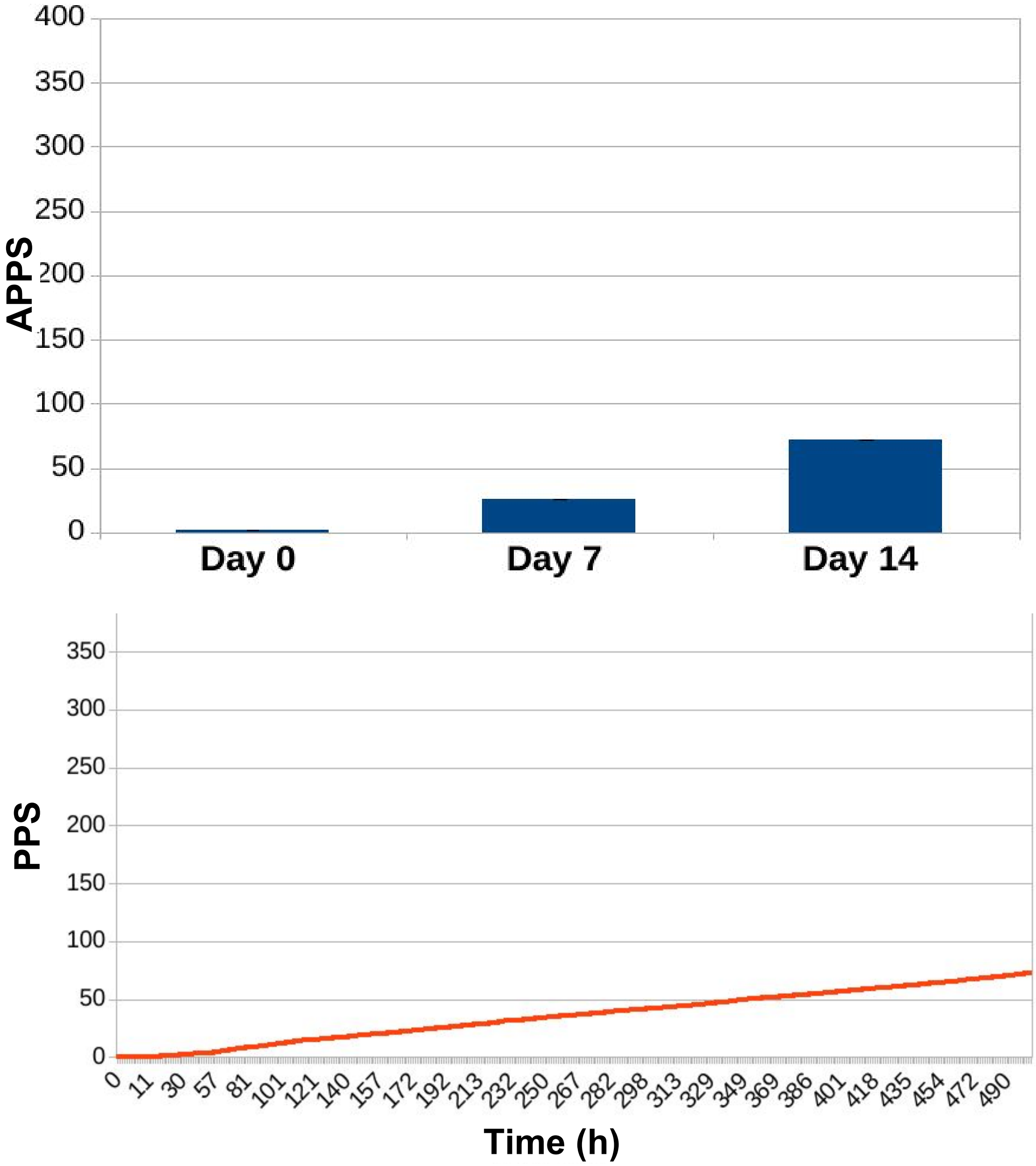


Figure 4.44. Results of PPS tracking for ED3 in case of no treatment administration to the murine microbiotas. When no treatment was administered to murine hosts, AP is 2.27 ±0.04 at day 0, 25.3±0.07 at day 7 and reaches 72.29±0.07 at the end of the experiment, at day 14. Simulation tracks show PPS increasing steadily and slowly compared to 4.45). Taken from Bardini, Di Carlo, et al., 2018.

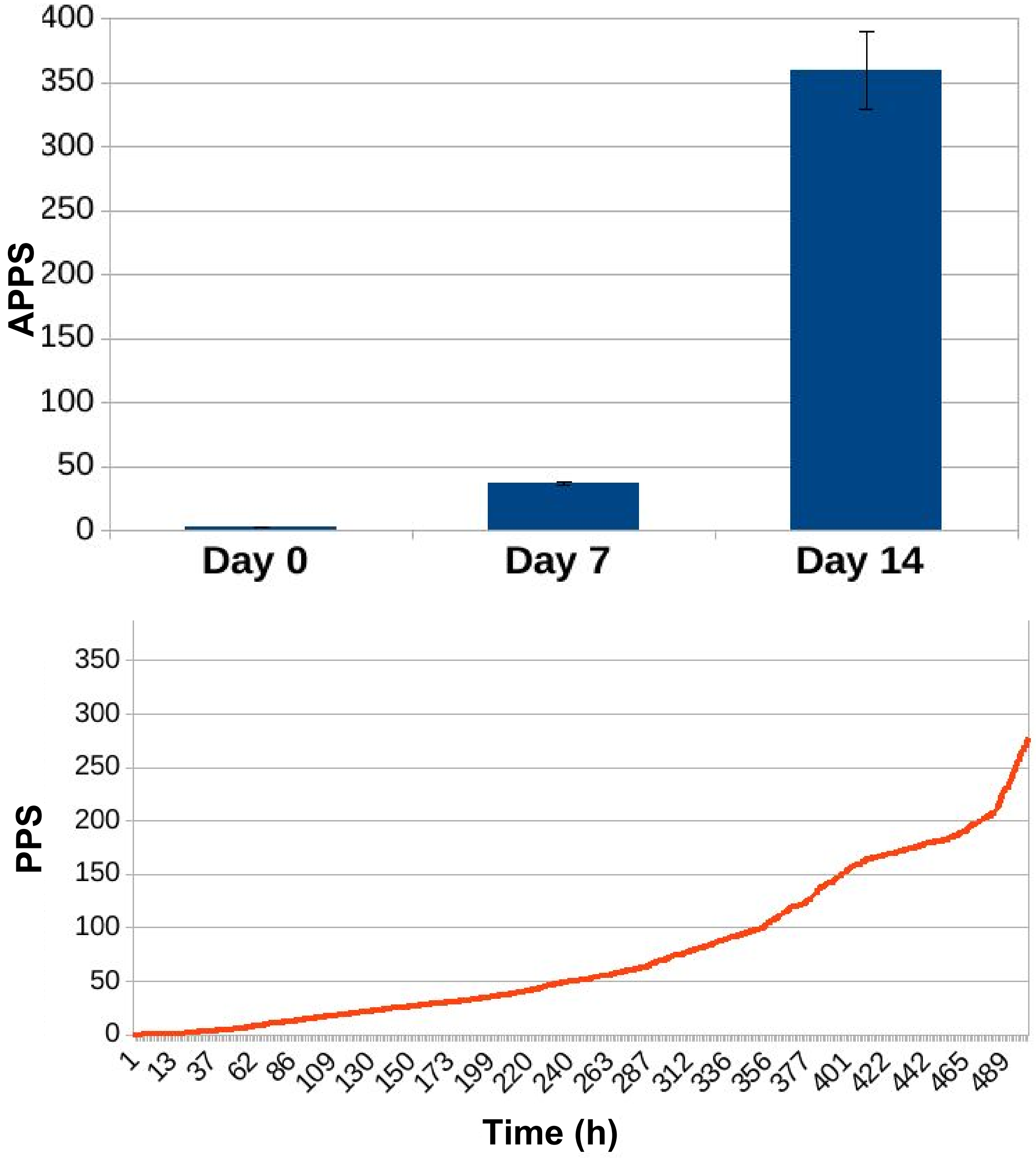


Figure 4.45. Results of PPS tracking for ED3 in case of daily administration of antibiotic treatment to the murine microbiotas. Simulating daily administration of antibiotic doses to the mice we observe, for the three relevant time points of the experiment, an AP of 2.5±0.05 (day 0), very close to the result in 4.44; after seven days of daily antibiotic administration, AP is 37.57±1.04 (day 7), and after seven days more reaches the value of 359.21±30.11 (day 14). The simulation curve shows how PPS grows faster compared to 4.44, reflecting the movement of population dynamics towards the establishment of the domination of the microbiota by resistant species. Taken from Bardini, Di Carlo, et al., 2018.

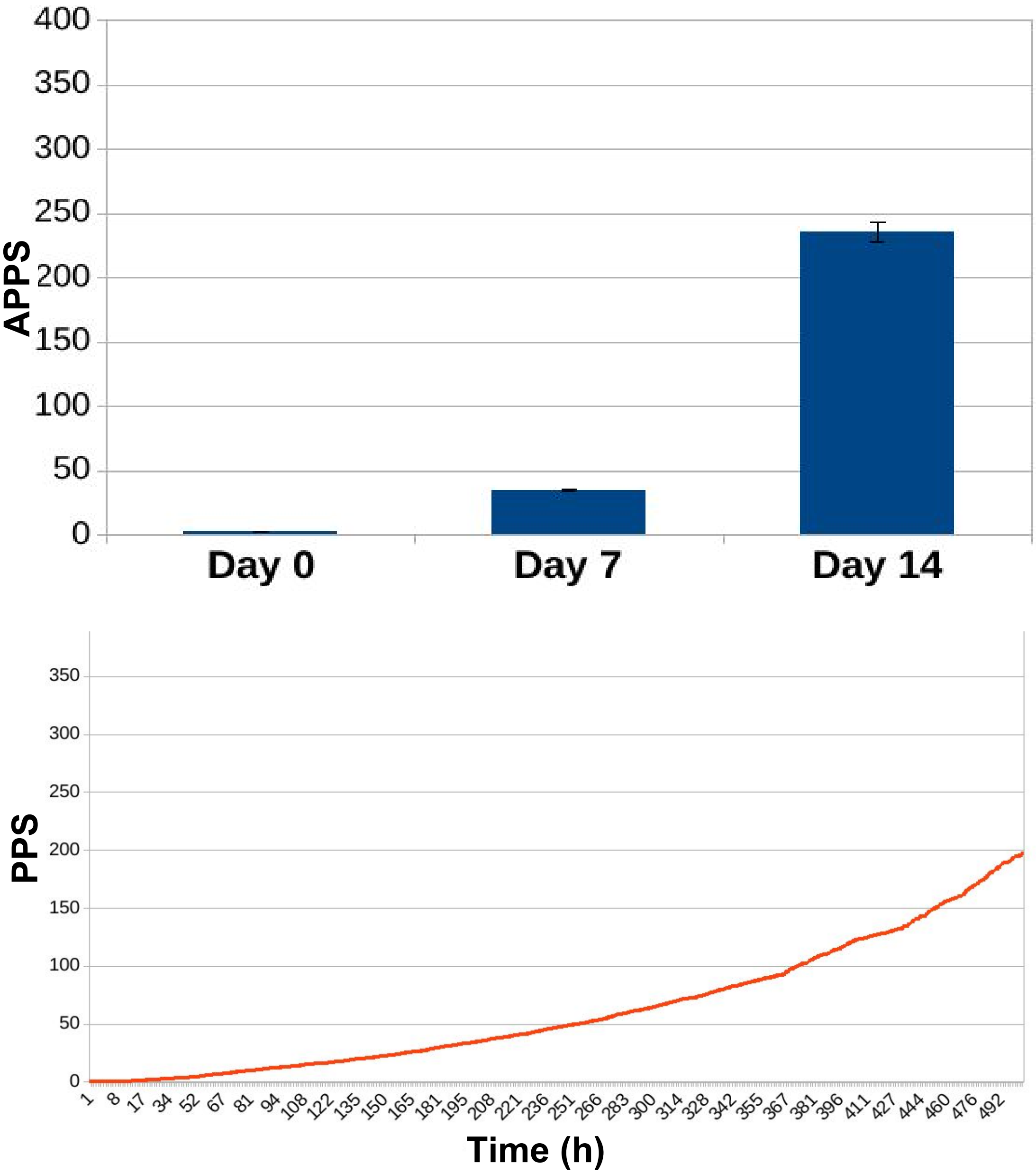


Figure 4.46. Results of PPS tracking for ED3 in the case of daily administration of antibiotic treatment combined with Bifidobacterium reintegration to the murine microbiotas. The combination of antibiotic administration and bacterial reintegration lowers AP values both at day 7 and 14. While AP at day 0 has a value of 2.54±0.04, similar to those reported in 4.44 and 4.45, after seven days of combined treatment daily administration AP is 35.25±0.51, and at day 14 reaches the value of 235.94±7.9. This result assesses the mitigating action by bacterial reintegration over the resistance levels observed when simulating the administration of antibiotic treatment alone 4.45. Simulation tracks show how PPS grows in time at a pace slower than that observed in case of antibiotic treatment alone(4.44). Taken from Bardini, Di Carlo, et al., 2018.

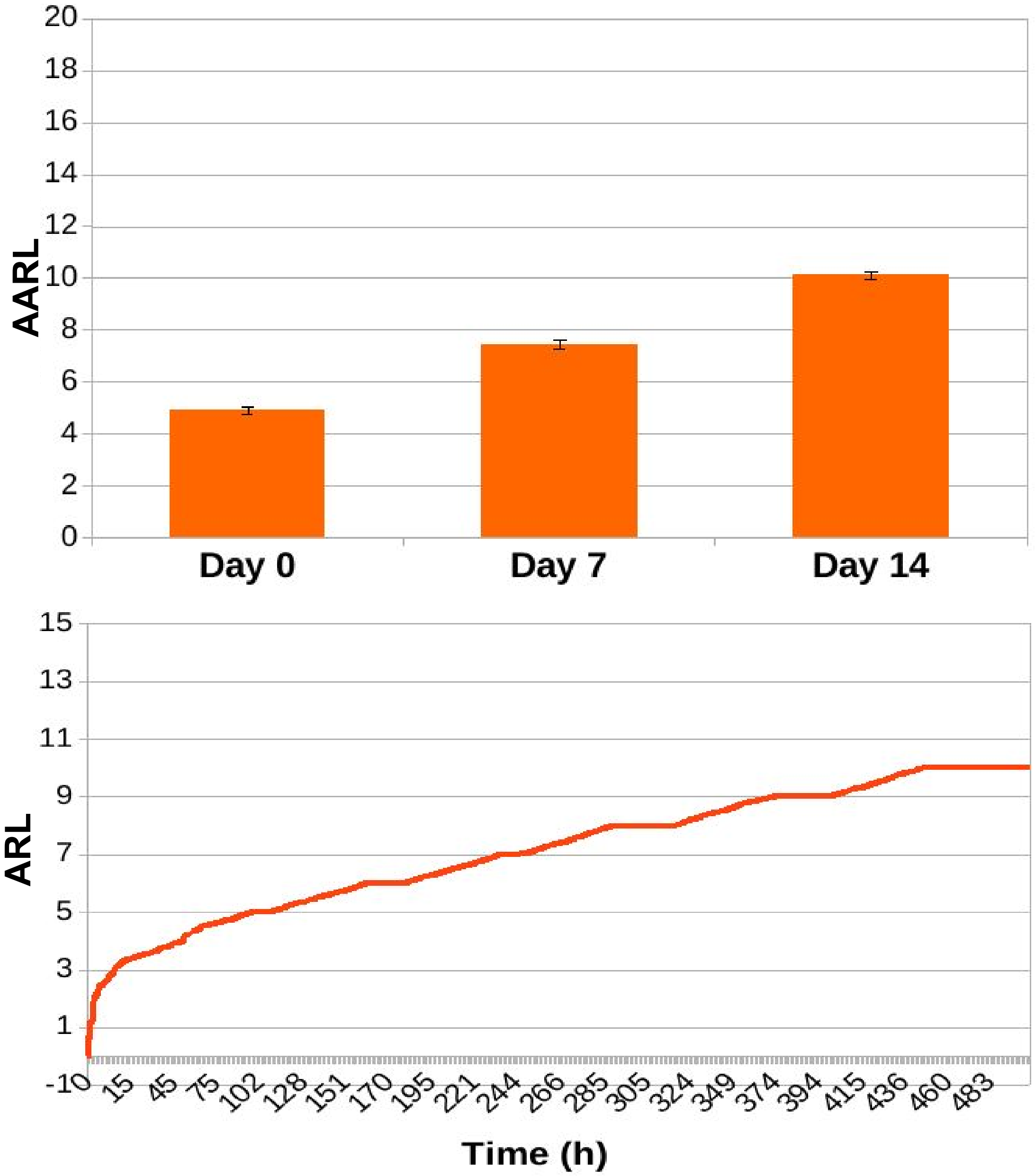


Figure 4.47. Results of ARL tracking for ED3 in case of no treatment administration to the murine microbiotas. When simulating a condition of absence of treatment, average ARL (AARL) assumes values of 4.83±0.12 at day 0, 7.6±0.19 at day 7 and 10.33±0.18 at the end of the experiment, at day 14. Simulation tracks show a slow and steady increase of ARL along simulation time. Taken from Bardini, Di Carlo, et al., 2018.

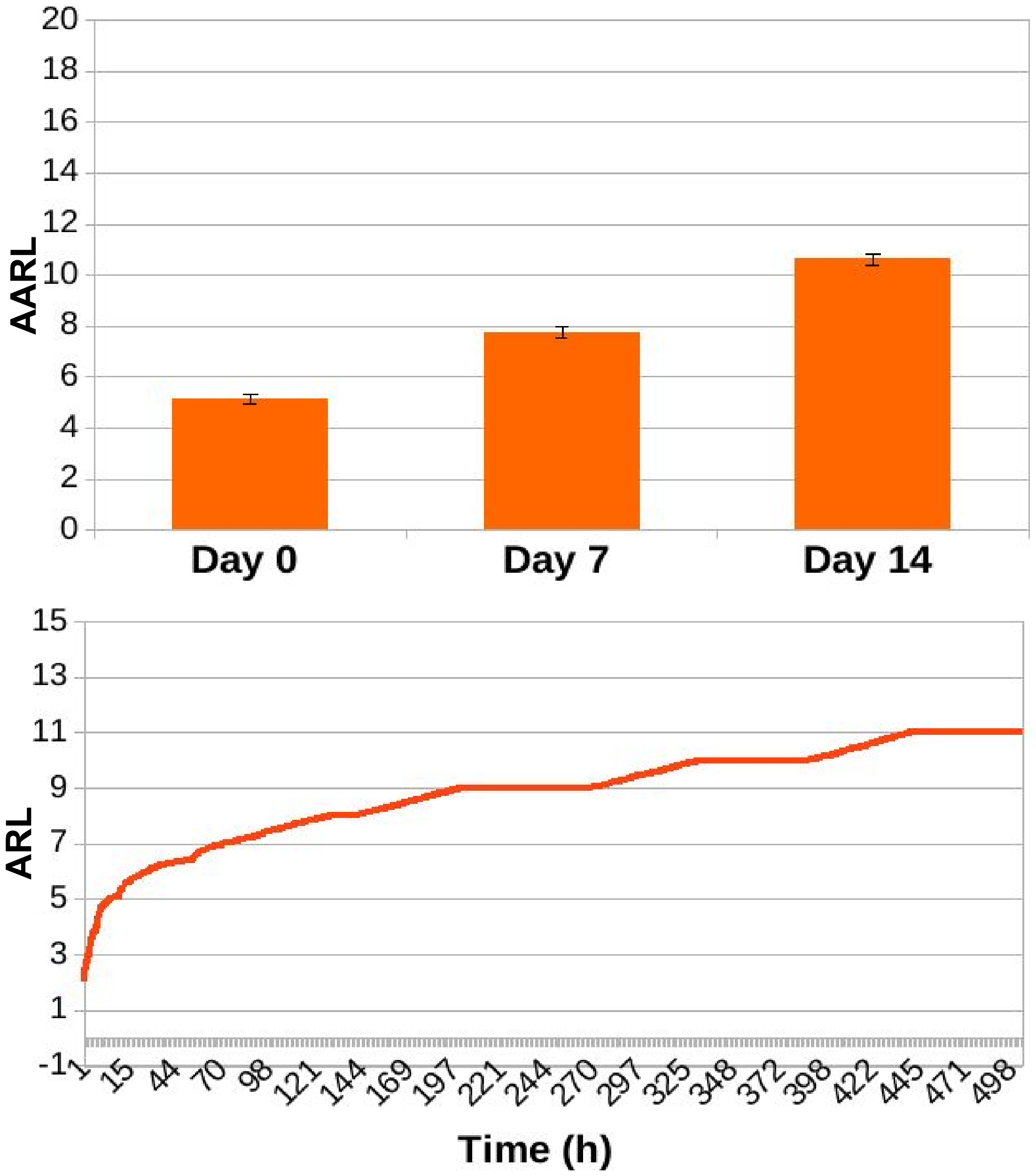
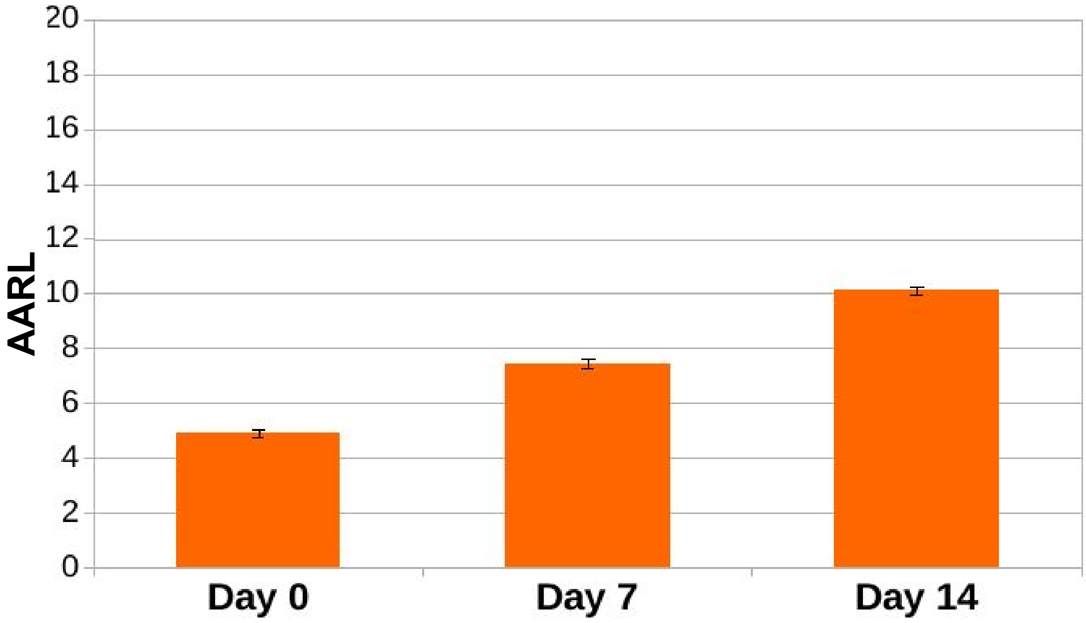


Figure 4.48. Results of ARL tracking for ED3 in case of daily administration of antibiotic treatment to the murine microbiotas. Simulation of daily antibiotics administration to the murine microbiotas yields to AARL values of 5.12±0.2 at day 0, 7.77±0.23 at day 7 and 10.61±0.22 at day 14. Similarly to 4.47, simulation tracks show a slow and steady increase of ARL along simulation time. Taken from Bardini, Di Carlo, et al., 2018.



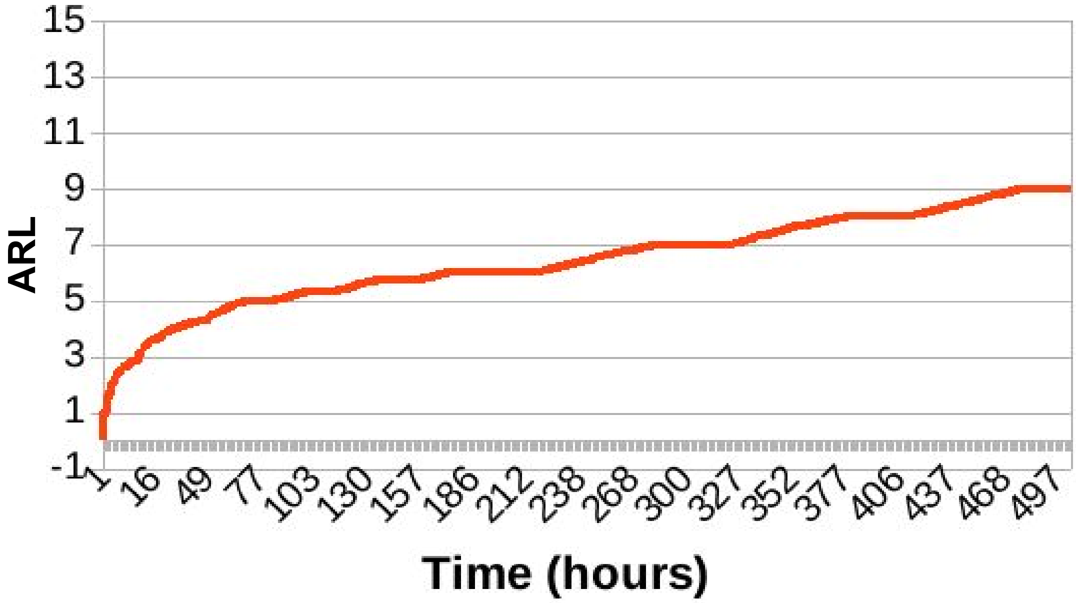


Figure 4.49. Results of ARL tracking for ED3 in case of daily administration of antibiotic treatment combined with Bifidobacterium reintegration to the murine microbiotas. Combining Bifidobacterium reintegration to antibiotic doses in daily treatment administration AARL is 4.93±0.14 at day 0, 7.46±0.16 at day 7, and 10.13±0.15 at day 14, and simulation tracks show a slow and steady increase of ARL along simulation time, similarly to what it is observed both in case of no treatment administration (4.47) and of daily antibiotic administration alone (4.48). Taken from Bardini, Di Carlo, et al., 2018.

4.7 Conclusions

This Chapter introduces the modeling approach developed for responding to requirements posed by systems biology. This approach mainly responds to those posed by system complexity. It recapitulates many strengths from the existing approaches from state of the art, both at the formalism and strategy level.

The Nets-within-nets formalism combines the computational advantages of discrete, agent-based and event-driven models with complete mathematical description and visual representation. Supporting all the features of other Petri Nets formalisms, NWNs have excellent expressiveness capability and support different levels of abstraction in the same model. Also, it allows specifying a boundless number of levels, and their interdependencies. NWNs models are equally suited for state-based and process-based representations.

Thanks to the capability to represent different types of information in the same model, the proposed approach supports integration processes combining different existing models or pieces of knowledge is a representation based on NWNs. The possibility of supporting model integration processes equals the capabilities of multi-level and hybrid models, in addition to maintaining formalism uniformity, facilitating subsequent formal analyses, and enabling model-based knowledge representation and exchange.

The proposed modeling approach leverages NWNs for supporting multi-level and multi-context architectures. That is, not only models represent multiple organization levels of the system. They support, in the same architecture, the explicit representation of multiple views over the same level. In particular, it is possible to describe a biological actor in reason of both its spatial context and its process context, at all model levels.

For these reasons, the approach flexibly applies to a variety of complex biological processes, modeling objectives and domains of systems biology. Initially, the proposed method develops around the requirements posed by ontogenetic processes, which add on the top of those relative to systems biology a much stronger need to explicitly model spatial and process organization of an evolving system. For this reason, the proposed models quickly adapt to applications posing a smaller set of requirements. This Chapter provides three application examples, each one including a statement of the approach to the scientific domain and the modeling objective, the structure of a core functional module for the topic of interest, and a specific case study or application example. The first example concerns ontogenesis aiming to gain new insights and predict system behavior after mutations, and treats patterning as a functional module, providing the applicative example of VPC specification in C. Elegans. The second example has synthetic biology as an application domain, aiming to support the design of complex synthetic biological systems, and has cellular communication between cultured cells as a functional module, which is applied, in a theoretical case-study fashion, to NO-based, EGFP-producing inducible communication between mammalian cells *in vitro*. The third example concerns epidemiology and aims to study the spread of antibiotic-induced antibiotic resistance in the microbiota and to support the design of innovative treatment protocols able to prevent it. The central functional module in this example covers HGT mechanisms in the microbiota, and the overall model also considers bacterial predation and MDR in the murine microbiota, integrating mathematical descriptions from two experimental works. This model mainly relies on the representation of states, rather than providing spatial aspects, showing how the proposed modeling approach is easily adaptable not only in representing different types of information from the system, but also different organizations of knowledge. These kinds of choices are often determined by information availability, like in this case. When spatial information is available or required for the model, the proposed approach is ready to embed it, being the spatiality model intrinsic to the framework. This example also shows how the presented approach not only can serve the scope of investigating causal, quantitative relations between different events, but they can also function as supports for decision making processes and development of clinical strategies. Also, this application shows how the integration of structured information from different sources can maintain formalism uniformity.

The presented approach relies on the specification of building blocks, recapitulating the concept of meta-transition Bonzanni, Krepska, et al., 2009, which in turn compose functional modules. Each of these substructures is multi-level and multi-context. Also, thanks to formalism uniformity, compositional processes based on functional modules are then inherently supporting cohesiveness and consistency in model construction. Also, this provides a form of modularity and allows for the direct porting of building blocks or functional modules from an application to another one, as shown in the presented examples. Portable representations make knowledge portable, *via* either entire models, building blocks or functional modules. Thanks to the capability for flexible abstractions of the presented approach, it is possible to exchange more models in their most abstract form, recapitulating more of a modeling approach to the general biological problem, and then tuning architecture, and the parameters of building blocks or functional modules to the specific application case. In this way, the modeler retains more representational flexibility than if a direct model modification occurs.

In this way, the proposed approach also responds to some level to requirements relative to knowledge representation and exchange. It does so only for users able to manipulate the underlying modeling formalism. The semantics of each building block can only emerge from knowledge stored in the model itself, which is self-explanatory for them. To adequately respond to the requirements relative to systems biology as a research domain, it is necessary to take into account its inherent diversity. Many different actors populate this rich research domain, and some have a purely experimental background, and no education in computer science or modeling. Adequate knowledge exchange is necessary to enable the effective collaboration between interdisciplinary profiles in systems biology. Accessible representations make knowledge effectively exchangeable. Moreover, they can make non-expert users able to leverage the inference capabilities of computational models.

To proceed in this direction, in the following Chapter a high-level model description language for the presented modeling approach is introduced.

Chapter 5

A model description language for systems biology

5.1 Short summary

To make multi-level and multi-context NWNs models accessible for non-expert users as well, the Biological System Description Language (BiSDL) has a high-level syntax recapitulating the domain-specific language of experimental biologists. At the same time, it also covers the low-level formalism elements. Also, BiSDL supports modularity: a description can make use of other descriptions, representing interconnected and nested models. The expert user can build up models under the multi-level, multi-context approach using BiSDL, creating re-usable modules corresponding to biological structures and processes. They can store these modules in libraries. Non-expert users can access libraries and access the knowledge stored in existing modules, as well as re-use, customize and combine them into high-level models by merely connecting them, and tuning their parameters. A custom compiler generates NWNs models from BiSDL descriptions, and a custom simulator directly simulates them. In this way, system dynamics is accessible as well to the non-expert user.

5.2 Introduction

A diversity of professional profiles contributes to systems biology as a research domain. Computational biologists easily apply and develop computational tools to tackle biological complexity. On the other hand, pure experimentalists have limits in accessing and using these tools, because their expertise lies in different technical and scientific domains than computer science. An interdisciplinary field like systems biology grows out of diversity: different perspectives, information, and capabilities are indispensable for the field to effectively advance. The necessity to support exchanges implies knowledge representation tools respond to the needs of such a diverse user base. In other words, they should function as enablers, making a large amount of expert biological knowledge and the capabilities of computational models accessible for the broader range of profiles possible, including pure computer scientists and pure experimentalists.

Under this perspective, when functioning as knowledge bases, models must become accessible to non-expert users as well, so they can get the information models to carry, and contribute to them with their own brand new expert knowledge.

Like in other technological and scientific domains, in systems biology high-level, domain-specific languages have the objective to make formal representations manageable for non-expert users as well. The strategy they rely on is the creation of an intermediation layer between the user and the low-level formalism of the model. This layer needs to be both user-readable and machine-ready.

The multi-level, multi-context modeling approach presented in the previous Chapter, relying on the Nets-within-Nets formalism, targets complex biological systems expressing hierarchy, encapsulation, selective communication, spatiality, quantitative mechanisms, and stochasticity. The modeling approach alone aims to target systems biology requirements on the front of knowledge inference and representation, but it satisfies them for a subpart of the potential user base in systems biology: non-expert users are not familiar with these kinds of supports. This lack of competence impedes knowledge exchange processes between expert users and them, which would be beneficial for both and systems biology in general. This Chapter introduces a model description language able to represent knowledge contained in the NWN models in a way that makes it accessible and usable for non-expert users as well.

The deep insights required for modeling biological complexity require a holistic perspective over biological processes. That is, it is necessary to comprise multiple system levels and different types of information in models, as well as to the choice of the level of abstraction in different model parts. This choice affects the accuracy and expressivity of the model, as well as the computational power required for simulating it. While High-Performance-Computing (HPC) platforms can face the computational requirements implied, a different shade of complexity emerges on the representational side. Systems biology shows remarkable diversity not only receiving the contributions from very distant professional profiles, such as a computer scientist and an experimentalist. Diversity concerns experimental biology itself as well. Biology as a research domain shows high compartmentalization but, contrarily to biological membranes, boundaries are not that semi-permeable. The different sub-domains, individually, can only describe portions of the overall system. Each of these sub-domains has specific scopes when investigating biological complexity, and particular history, culture, and representational styles. This fragmentation poses a significant challenge to modeling languages. First of all, it is necessary to understand the relationships between different sub-domains and their respective scientific outputs in terms of knowledge generation. Then, this higher-level understanding must make different information flow into a model using a uniform description language.

At the moment, communications between different sub-domains still need to improve, and this reflects in the existence of a variety of description languages for models of biological complexity: each one somehow relates to the representational style from a sub-domain of biology.

The proposed language aims to recapitulate the representational capabilities of existing domain-specific languages in systems biology, providing a consistent representational style to facilitate knowledge representation and exchange. For a more detailed introduction to the language see Muggianu et al., 2018.

Summing up, the presented model description language intends to serve two different purposes:

1. *to be biologist-friendly*, human-readable, able to model both the behavior and the structure of a biological system, able to support several biological sub-domains, and flexible and modular enough to be used at different levels of abstraction;
2. *to be computation-ready*, supporting the simulation of the modeled system. For this reason, BiSDL design revolves around the syntax of the Nets-within-Nets formalism, and BiSDL descriptions automatically translate into a fully executable NWN model.

Similarly to what already presented for the modeling approach, the fact the ultimate target process for the design of BiSDL was ontogenesis does not impede the flexible application of the the language to other biological phenomena. In fact, in its final form, the BiSDL is intended to be used to model a wide spectrum of biological systems.

In structuring the language, the VHSIC Hardware Description Language (VHDL), a very well-known language used to model complex digital circuits and systems Perry, 1993 inspired the design and syntax.

5.3 Existing model description languages in systems biology

The *COmputational Modelling in BIology NEtwork* (COMBINE) intends to coordinate the ideation and development of standards for models of biology COMBINE, 2018. To this aim, in the context of this initiative, different existing languages, specific to many sub-domains of biology, have been analyzed and mapped. Figure 5.1 summarizes the ones found to be most relevant according to the COMBINE initiative.

This section provides an overview of the existing formats for describing models in biology.

Languages for data exchange The BioPAX Data Exchange format Goldberg et al., 2010 intends to support cross-database data exchange, making pathway data substantially more comfortable to collect, index, interpret and share. The Synthetic Biology Open Language (SBOL) R. S. Cox et al., 2018; Galdzicki et al., 2014 supports the exchange of information about structural and functional aspects of biological designs to embed into synthetic systems.

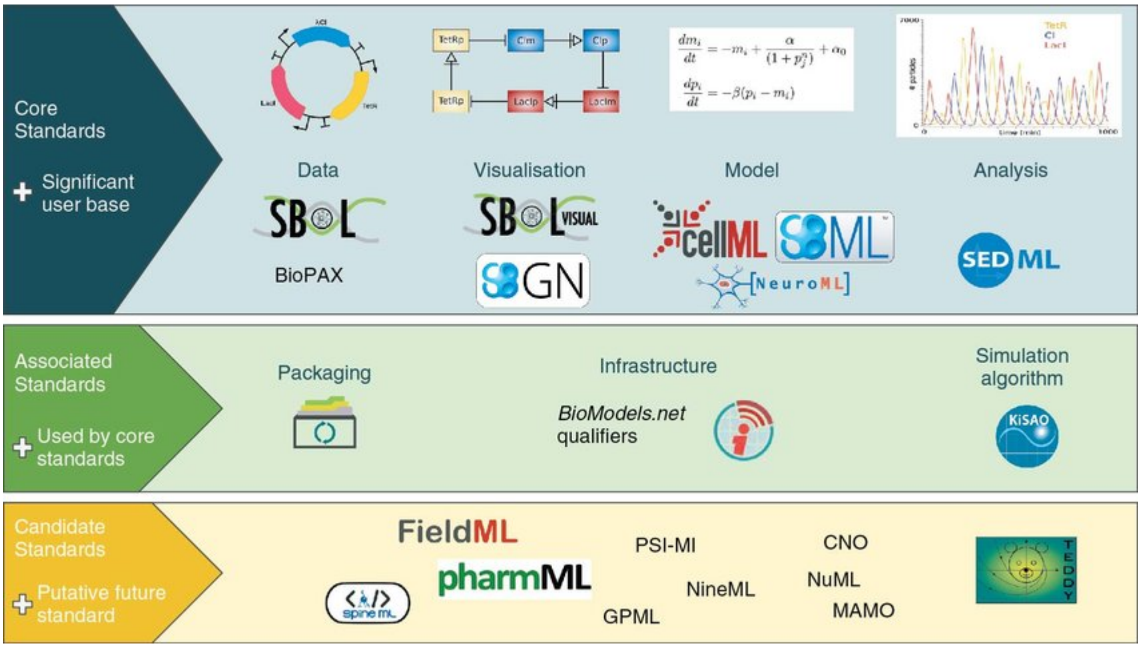


Figure 5.1. Overview of existing standardization efforts in Systems and Synthetic Biology covered by COMBINE (from Schreiber et al., 2015). Taken from Muggianu et al., 2018.

Languages for visualization The Systems Biology Graphical Notation (SBGN) Le Novere et al., 2009 describes in clear and visual way signaling pathways, metabolic networks, and gene regulatory networks.

Languages for model exchange Exchanging data or visual representations pose an easier challenge, compared to exchanging complete models. Under the knowledge exchange perspective, the model description is one of the most delicate phases: it should support the development of reusable and easily integrated model architectures. At the moment though models rely on different, often non-interoperable formalisms, and include references to databases or ontologies not linked between each other. Also, knowledge exchange should rely on good practices on how to generate and summarize simulation outcomes, so to easily share that as well.

The COMBINE initiative currently foresees four languages as virtuous formats for models of biology.

System Biology Markup Language (SBML) As in Chaouiya et al., 2013, SBML is an XML-based machine-readable format for representing models. In SBML, biological entities undergo modifications by processes that occur over time. The SBML has, as frequent targets, biochemical models. Its syntax tightly links to the scopes of this sub-domain, and the underlying mathematical formalism of choice: systems of differential equations.

CellML language As in Lloyd et al., 2004, CellML is an XML-based language including information about model structure, equations describing the underlying processes, and metadata about the model facilitating model storage and exchange. CellML is not related to a specific domain within biology, supporting the description of a wide range of systems. Moreover, it is flexible enough to comprise newly discovered mechanisms.

NeuroML As in Gleeson et al., 2010, NeuroML is, in the first place, a collaborative initiative to develop an XML-based description format for defining and exchanging descriptions of neuronal cells and network models. NeuroML is the language with the highest sub-domain specificity among those considered. The project also devises a simulation engine. At the aim to improve NeuroML in the description of neuronal models, and in particular synapses, developers designed *LEMS* Cannon et al., 2014, a *Low Entropy Model Specification*. LEMS supports a compact, non-redundant, human-readable, human-writable and declarative model description style for biology. It is too generic to count as a domain-specific language. References to biological concepts or structures are not mandatory for the user.

Languages for exchange of simulation outcomes SED-ML Kohn and Le Novere, 2008 encodes the description of simulation experiments in XML, categorizing types of experiments that it is possible to perform.

In Yoshiyuki Asai, Abe, et al., 2014, authors describe the integration of two modeling mark-up languages. Their platform combines representational capabilities, usability, and interoperability of two existing formats: the Systems Biology Markup Language (SMBL) and the Physiological Hierarchy Markup Language (PHML).

SBML Hucka et al., 2003 targets sub-cellular mechanisms relying on ODEs, while PHML suits hierarchical systems Yoshiyuki Asai, Y. Suzuki, et al., 2008. In this work, a PHML framework embeds SBML modules, supporting the representation of several system organization levels in a single description. When different modules communicate, "get" or "set" functions consistently handle values from or to communicating modules. This approach carries the need to perform consistency checks over the resulting model, which can show high computational complexity.

5.3.1 Limitations of available languages

All analyzed languages share to some extent the ability to describe entities, processes, and communications between entities or between entities and processes. Nevertheless, none of them has all these characteristics, which are required to describe models of complex and generic biological processes, integrated into a single language.

The target of the presented modeling approach is complex biological processes, such as ontogenesis. A model description language needs to express all of the features the NWN formalism and the modeling approach to these processes have, making them accessible to the non-expert user as well.

This challenges existing formats, starting from the requirement to express spatiality and mobility. Only CellML, thanks to FieldML, and NeuroML that can describe the spatiality in the neurological field can express spatial information. Hierarchical organization as well and the underlying communication mechanisms in biological systems challenge current description languages, which tend to provide models with a flat structure, except for NeuroML, which is still too specific to the domain of the nervous system.

Table 5.1 recapitulates essential features of the formats considered, to highlight their strengths and limitations considering the scope of the presented modeling approach.

5.4 Exploring biological semantics

What determines the effectiveness of a domain-specific language is its capability to support the natural way the user thinks. At the aim of adapting the language semantics to the real needs of non-expert users, that is, experimental biologists, the necessity emerges to map the actual way they represent knowledge through natural language. Two strategies were put in place to investigate these aspects.

- qualitative user interviews with experimentalists;
- concept mining over existing ontologies for biological terms.

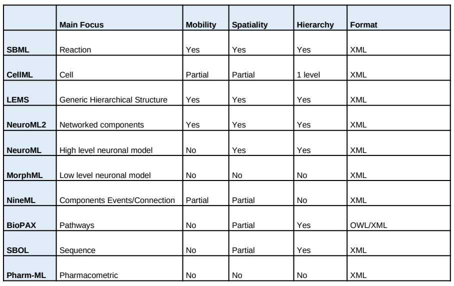
While the main advantage obtained from the interviews was collecting hints and caveats to orient the design process, by performing concept mining more quantitative insights emerged, which guided the prioritization of biological concepts to cover with the language semantics necessarily.

While the instrumental part of this work mainly involves text mining tasks, its real aim is better defined by *concept* mining. That is, its objective is the extraction of concepts from biological ontologies. In the design of this study, a concept is intended as a set of co-occurring words in ontologies lemmas, directly referring to the main linguistic structures underlying knowledge representation in biology.

5.4.1 Co-occurrences networks

The first step to extract concepts from biological ontologies is to detect co-occurrent words among those populating them. When analyzing ontologies, parameters guiding the detection of co-occurrences vary from those used in mining longer texts. It is possible to interpret ontologies as sets of individual lemmas, i.e., sentences, or separate texts. Then, the computation of co-occurrences considers every single lemma as a text of reference, and in each of these texts, the maximum distance possible between words matches the length of the lemma itself, introducing specific caveats

Table 5.1. Available modeling languages in systems biology . Taken from Muggianu et al., 2018.



	Main Focus	Mobility	Spatiality	Hierarchy	Format
SBML	Reaction	Yes	Yes	Yes	XML
CellML	Cell	Partial	Partial	1 level	XML
LEMS	Generic Hierarchical Structure	Yes	Yes	Yes	XML
NeuroML2	Networked components	Yes	Yes	Yes	XML
NeuroML	High level neuronal model	No	Yes	Yes	XML
MorphML	Low level neuronal model	No	No	No	XML
NineML	Components Events/Connection	Partial	Partial	No	XML
BioPAX	Pathways	No	Partial	Yes	OWL/XML
SBOL	Sequence	No	Partial	Yes	XML
Pharm-ML	Pharmacometric	No	No	No	XML

in generating the co-occurrences matrix. For a more detailed discussion of the strategy underlying the text mining strategy implemented, see Muggianu et al., 2018.

It is possible to visualize the co-occurrences matrix as a network, and in this case, visualization relied on the Python library NetworkX Hagberg et al., 2008. In the network, weighted edges mark co-occurrences between pairs of words, and nodes represent individual words. Nodes size is directly proportional to the occurrence counts of the word they represent.

Figure 5.2 shows the results of network visualization over Cellular Component, a branch of the Gene Ontology.

The portion of the co-occurrences network shown displays relations between terms, highlighting differently colored clusters. These clusters, in the presented analysis, define *biological concepts* of interest.

Biological concepts, as defined in this way, guide the choice of the main structures in the language syntax: their design takes into account the concepts emerging from this study by making sure they can find proper representation.

This process relies on two main sources which guide the language design:

- Cell Behaviour Ontology instructed the syntactic structure;
- Gene Ontology provided the concept organization to embed in it, instructing the semantics of the language.

In particular, the structure of each description supported by the language devises:

METADATA This header section provides metadata about the model: creation date, names of the authors, and name of the model. This part enables effective model exchange and traceability, making it possible to keep track of modifications and actors involved in the model creation process.

MAIN In each description, the central section covers different abstraction levels. Starting from the higher-level of description:

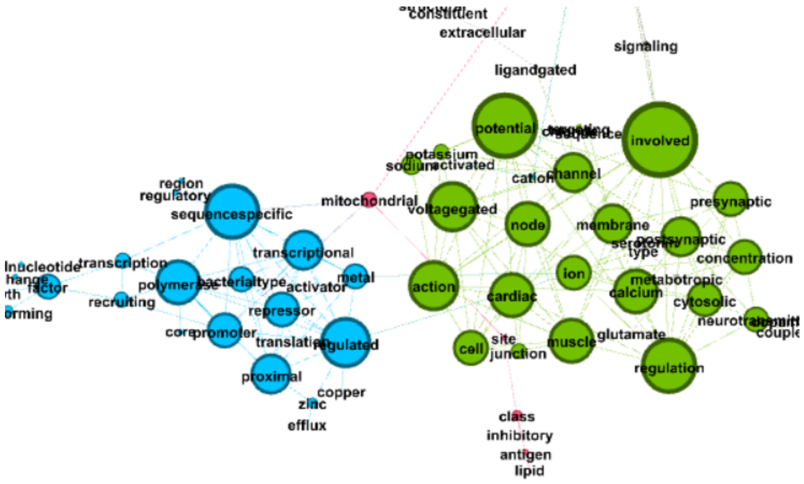


Figure 5.2. A detail from the co-occurences network obtained from the Molecular Function branch in Gene Ontology.

- the Environment recapitulates the most external component of the model. Its properties include a *name*, a *type* (chosen from a predefined pool) and spatial information about the system, such as its *dimensions* and *shape*. This section includes Entities and Processes:
  - each Entity, as a subcomponent of the Environment, participates in the description of the *structural* aspects in the system. Its properties include a *unique name*, and a *type* chosen among those proposed by the language. There are many types of entities, which differ by particular specifications, as better explained in the following sections. Yet, all of them have in common the specification of spatial properties: *shape*, *dimension* and *position* respect to the Environment.
  - each Process, as another kind of subcomponent of the Environment, covers a part of the behavioral aspects of the system: it describes either the functionality of an Entity or the relation between different Entities. Its properties include a *unique name*, and a *type* chosen among those proposed by the language.
- The Hierarchy of the system covers the definition of the internal architecture of Entities. Each Entity can function as an Environment itself: for this reason, each Entity requires the same specifications as those described above for the Environment. Processes as well can contain sub-Processes, following the specification scheme of the Process structure in turn.

This general structure, when employed for describing systems biology models, face the great diversity and vastness of the necessary vocabulary to include. Also, the inclusion of new concepts implies the re-implementation of a whole Hierarchy of syntactic terms. Compiling BiSDL descriptions into NWN models at this stage is computationally expensive, and the resulting compiled models, in this way are entirely opaque to users. Which satisfies the needs of non-expert users, but it fails to satosfy those of expert users. They may want to manipulate biological knowledge at all available levels of abstraction, from high-level descriptions of biological concepts to the NWN formalism elements at the lower, model level. Moreover, besides exchanging knowledge under this paradigm, expert users may also want to encode new concepts into the BiSDL *corpus*: they need to be able to modify structures at all existing levels.

The BiSDL structure evolves, following the intention to benefit both biologists with no expertise in computer science and experts in computational biology.

This level of flexibility implies the representational power of the language can range from low-level models implementations, using the NWNs syntax, to high-level descriptions, whose semantics ties to biological concepts. In other words, BiSDL intends to support two usage modes. On one side, non-expert users only perform model descriptions, manipulating high-level concepts. Their main benefit is the possibility to access rich and structured knowledge bases and to combine biological concepts and eventually to study their dynamics readily. On the other side, expert users manage both high-level descriptions and manipulation of lower-level functional blocks. They have an additional benefit, that is, thanks to their access to the low-level syntactic elements, they can create modules recapitulating biological knowledge from scratch, contributing to populate the libraries non-expert users rely on for accessing that knowledge. In other words, for non-experts BiSDL functions as a high-level, domain-specific model description language; for experts, it functions as a flexible modeling formalism. It allows manipulating syntactic structures at different levels of abstraction *from the model*, from NWN formalism elements to BiSDL modules.

5.5 Sources of inspiration

BiSDL design takes inspiration from other endeavors on the front of domain-specific languages. Many other domains present representational challenges, due to the complexity of the systems they intend to treat. In particular, the domain of VHSIC (Very High-Speed Integrated Circuits) design provides good inspiration. In this field, the Gajski-Kuhn Chart, or Y-Chart Gajski and Kuhn, 1983, depicts the different perspectives relevant to the design process.

Gajski-Kuhn Y-chart

In the VHSIC domain, Very-Large-Scale Integration (VLSI) is the process of creating an Integrated Circuit (IC) by combining hundreds of thousands of elements into a single chip. The Y-Chart summarizes the good practice existing in VLSI which devises the separation of the *usage* model from *architectural* and *implementation* details. also, it is required to find a functional mapping between these three aspects.

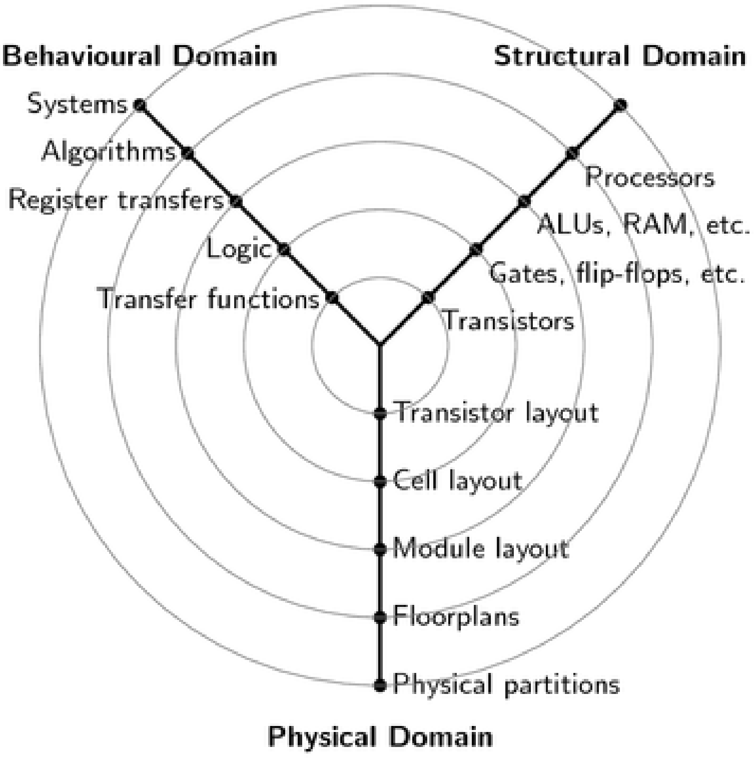


Figure 5.3. The Gajski-Kuhn Y-chart.

As in Figure 5.3, three radial axes recapitulate the domains of interest. Each domain has different levels of abstraction, represented by concentric rings.

The most external level represents the overall *architecture* of the chip, and the closer a level is to the center, the more it includes implementation details.

The three axes cover the different aspects of interest, at all available levels of abstraction:

- the *behavioral* domain describes the temporal and functional behavior of the system;
- the *structural* domain describes subsystems and how they intertwine to form the overall system;
- the *physical* domain describes the geometry of the system and its sub-components, as well as their size, shape, and position in space.

Intersections between concentric rings and radial axes point to different aspects, considered at the different abstraction levels. For a more thorough description of the behavioral aspects at different abstraction levels refer to Muggianu et al., 2018.

VHDL

VHDL (VHSIC Hardware Description Language) Shahdad et al., 1985 is a domain-specific language to describe hardware systems designed with the aid of the Gajski-Kuhn Y chart. As the Chart, VHDL can manage different levels of abstraction in describing systems, with a focus on the structural and behavioral domains.

The Design Entity is the minimal unit in VHDL usage, and it describes hardware components as black boxes. That is, it explicitly models only the interfaces they exhibit towards the external environment. The Design Entity applies to different hierarchical levels of the system. For example, it can describe a logic gate, as well as an entire system, implying links to architectures of different kinds, which in this context recapitulate three different views over the system:

- *Dataflow* specifies logic expressions describing how elements of the language interconnect on the informational plane, including block declarations and instantiations, necessary procedures and timing rules for execution.
- *Structural* specifies the interconnection of components at the functional level, that is, through the signals they exchange.
- *Behavioural* describes the algorithmic functioning implementing system behavior, and follows a sequential logic in terms of instruction execution, simulating circuit parts at a higher abstraction level than that of specific components.

5.6 Biological Systems Description Language

As discussed in the introduction, BiSDL needs to be biologist-friendly and computation-ready. On the one hand, this means that some features of the language must link to the NWN formalism used to simulate the described system. On the other hand, its syntax must be able to hide the technicalities of the final implementation, allowing a focus on the description of the actual biological system. The design process started from the examples provided by the Y-Chart and the VHDL and adapted the principles underlying these existing tools to the challenge of making NWN models accessible to both expert and non-expert users.

A Y-Chart for BiSDL

The Gajski-Kuhn Chart develops for a domain belonging to the field of engineering. More specifically, it dedicates to top-down design processes, in which the designer has, in principle, total control over function implementation through manipulation of system components. For this reason, it is intended to cover all components the designer may use. Each sub-component is a known entity in the physical system, and its behavior is possible to identify and model in an accurate way. COmplexity, in this case, emerges from a large number of components from the system, and their intertwinings and interconnections. On the other hand, dealing with biological complexity a top-down approach is not feasible, at least in the way hardware engineering originally conceived it. System biology models necessarily perform approximations and omissions, mainly due to the lack of knowledge about biological complexity. For a more thorough discussion of the relationship between life sciences and engineering, see Section 2.2.4.

Considering this premise, the Y-Chart has many interesting characteristics able to guide endeavors in computational systems biology. For example, the Physical domain can respond to the requirements concerning spatiality biological complexity poses to modeling formalisms. The different abstraction levels in the Y-Chart reflect the original intention to support the manipulation of BiSDL models at higher and lower levels.

At the aim to adapt the Y-Chart to the biological semantics, it is necessary to implement some modifications. This Section provides a detailed description of the BiSDL Y-Chart, as it constitutes the nucleus of the language and supports the following descriptions. Figure 5.4 recapitulates the presented scheme.

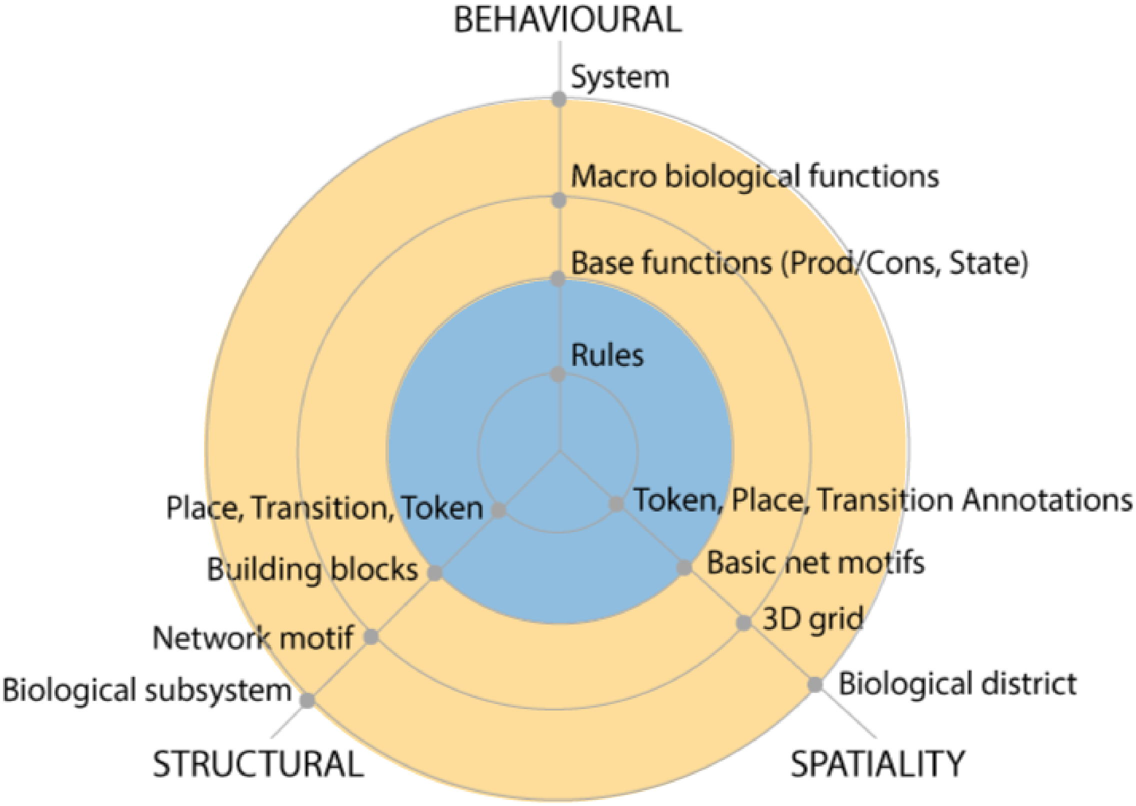


Figure 5.4. The BiSDL Y-Chart.

The three radial axes refer to the domains to cover in the description of biological systems:

- The Structural domain focuses on relations and communications between subparts. Different entities compose a model, simple or complex, and combine. This axis specifies the different abstractions performed on biological structures and how BiSDL manages them.
- The Behavioral domain describes the system focusing on processes and behaviors, including processes and functional activations taking place during the execution of the model.
- The Spatial domain explicitly describes spatial aspects of subsystems and their interrelations. Not only it covers geometrical aspects such as the shape and the dimension of system elements. It also describes their absolute and relative positions and directional movement.

While in the VHDL Y-Chart concentric circles refer to different levels of abstraction from the physical hardware system, in BiSDL they map to different abstractions *from the NWN model* to describe, which deals with abstraction from the physical system in turn, as better explored in the previous Chapter. These two perspectives collapse together when BiSDL serves the direct manipulation of elements from the NWNs formalism in model construction processes.

It is possible to observe the different levels in the Chart acquire different meanings at the three intersections with the axes.

Ist level At the most inner circle, low-level NWNs formalism elements live. Places and their interconnections describe structural and possibly spatial aspects; transitions and their local rules describe the behavior of the system during execution.

IInd level At this level, different NWN elements combine and form composite network motifs. This level links to more complex representations for more extensive substructures in the model and the interconnections between them, as well as for more detailed spatial descriptions. These structures follow the definition of meta-transitions Bonzanni, Krepska, et al., 2009 on the behavioral plane; that is, they are network structures corresponding to an identifiable complex process in the model. At this level information still has a general semantics, and does not link to any particular domain.

IIIrd level One level higher, network motifs acquire biological semantics, providing descriptions of biological processes (Behavioral domain), biological structures and their functional relations (Structural domain) and spatial aspects to this structures (Spatiality domain). Biological Building Blocks as described in Section 4.3 link to this level.

IVth level At the higher abstraction level, the NWN formalism underlying BiSDL models is completely opaque. Entities center primarily over biological semantics: complex biological structures descriptions combine different Building Blocks and interconnect with each other. Complex, yet identifiable biological processes emerge from them, and their spatial features are modeled in their entirety by detailed spatial descriptions. Complex Functional Modules from Section 4.3 relate to this level.

A description style for BiSDL

Considering the adaptations of the Y-Chart to the purposes of computational biology, this section describes how the VHDL worked as a starting point in ideating the BiSDL description style.

VHDL inspires BiSDL mainly respect to three main characteristics:

- Different description domains
- Modularity
- Design Entity as a black box

Different description domains BiSDL explicitly cover Structural, Behavioral and Spatial aspects of the described system.

Modularity Systems biology tends to represent biological complexity combining holistic and reductionist approaches. In fact, on one side it is good to have comprehensive representations of the whole system, its subparts and their interconnections, from which system behavior emerges. On the other hand, there is also the tendency to refer to precise functional building blocks that it is possible to associate to known behaviors, so to deal with at least partly understandable structures. These structures, on the representational side, correspond to modules maintaining their identity across models, and combine to form more complex structures. For this reason, Modularity is a central feature for a model description language. For the language, modularity not only is natural to the current representation of biological complexity, but it also enables the re-use of existing modules in different models. Also, once the boundaries of a module are defined, it is possible to modify it, creating different versions of the same module. In BiSDL, each module has Behavioral, Structural and Spatial aspects.

Design Entity as a black box In BiSDL, Modules are the fundamental Design Entities. A Module can be treated as a black box, making explicit only their interfaces and set-up parameters, mediating their connections to other Modules. Connections between Modules require either the fact they share a sub-Module or the use of a particular Entity, that is, the Channel, implementing synchronous communication mechanisms. To be unambiguously identified with a specific biological element, Modules need to carry a reference to a well-known knowledge base. This expedient facilitates reliable knowledge exchange, as well as the easy inclusion of Modules in existing models. Libraries can store existing Modules, allowing the non-expert users as well to re-use them. From a biological point of view, a module could correspond to any biological entity or process at any hierarchical level. For example, a module can describe a simple protein, a gene, a complete pathway, a metabolic subnetwork, an entire cell, or a biological process (e.g., gene transcription, functional activation of a protein or its degradation).

The NWNs modeling approach presented includes the concepts of Building Block and Functional Module already. Modules easily map to these constructs, making them easier to modify or merely combine and re-use.

All of these aspects reflect into the general template, which both expert and non-expert users can use to produce BiSDL specifications.

5.6.1 The general template

In BiSDL descriptions, all Modules use the general template depicted in Figure 5.5.

```
PACKAGE <package1>
...
PACKAGE <packageN>

ONTOLOGY <ONTOLOGY_NAME>=<url>
...
ONTOLOGY <ONTOLOGY_NAME>=<url>

MODULE <name> (<type> <nameparam>,.....)

 BIOLOGICAL REFERENCE
   <ONTOLOGY_NAME>.<ID_WITHIN_THE_ONTOLOGY>

 ENTITIES
   PLACE <name_place>,....
   ENTITY <name_entity>,....
   CHANNEL <name_channel>,....

 INIT
   <place_name>.attribute(<value>)
   <entity_name>.attribute(<value>)
   <entity_name> = <module_entity_name>(<params>,..)

 PROCESSES
   process(   keyword:entities_declaration,
       keyword:entities_declaration,
       ...,
       {transition_function},
       delay(N)
   )

   <module_process_name>(<param>,....)
END
```

Figure 5.5. BiSDL general template. Taken from Muggianu et al., 2018.

Following the structure of the general template, this section expands on the features supported by BiSDL.

Libraries of modules Since the general idea is to have a language that allows the description of a biological system in a way that is both human-readable and ready to be translated into an NWNs model, it is necessary to support both Modules creation and simple re-use. *BiSDL libraries* are supposed to store existing Modules, which can then be used to create BiSDL descriptions. The BiSDL itself does not define official libraries: it just supports modularity so the users can populate and rely on them.

Figure 5.5 shows the general BiSDL template. BiSDL descriptions start with the PACKAGE declarations, which can specify what to include from libraries.

Relevant ontologies The ONTOLOGY directives are used to unambiguously associate Modules with known biological entities. To do this, BiSDL requires specification of a set of URLs to recognized sources of knowledge (e.g., Gene Ontology Consortium, 2018, Pathway Ontology BioPortal, 2018).

Tunable Module parameters Following is the MODULE section, which contains the actual description of the module. A Module name must be unique inside the package containing it. Optionally, the Module declaration contains the respective parameters, whose value is specified at instantiation.

Before describing the core structure and behavior of the module, BiSDL requires specification of its BIOLOGICAL REFERENCE, i.e., the unique link to an element of one of the ontologies declared in the ONTOLOGY section. This is performed according to the following syntax:

```
<ONTOLOGY_NAME>.<ID_WITHIN_THE_ONTOLOGY>
```

Inside each MODULE, BiSDL allows the description of the modeled biological entity under the three domains from the Y-Chart.

Usage of modules In this part, the biological identification of Modules directs their functionalities. The ENTITIES sub-section instantiates the elements that the module will use accordingly. The interaction among Entities describes the behavior of the module. Each Entity will be linked to its implementation (MODULE) in the INIT section.

In BiSDL, there are three types of ENTITIES:

- ENTITY: a complete Module, implemented following the usage rules described in this section, declared in the PACKAGE section and drawn from libraries;
- PLACE: a place intended under the NWNs formalism;
- CHANNEL: a component specifically intended for synchronous communication between different Modules.

In an ideal hierarchical BiSDL description, only the lowest level modules will explicitly make use of the NWNs elements, and the higher levels support a high-level biological semantics.

The INIT section initializes each instantiated ENTITY, declaring its actual origin within one of the included libraries, and setting, if present, the required parameters. In the case of a PLACE, the INIT section is used to set a name, the initial token marking (the type of token(s), their value, and their quantity), and the ontology ID linked to it. Tokens can be of type int, float, double, string, black token (no type), or a complete Module (user-defined or taken from one of the included libraries). Different operations can target different token types. For example, for the numerical types all the basic mathematical operations are allowed (sum, difference, division, multiplication). The syntax to assign tokens to a place is the following.

```
place_name.token(token type * N)
```

N is the number of tokens.

The INIT section also sets the relative speed at which the Entity must be simulated, considering relative time scales in the overall model hierarchy. Each speed is a multiple of the minimal time unit in the simulation, which corresponds to the speed of the fastest module.

Both ENTITY and PLACE modules have a location within the BiSDL 3D grid attached. If exact coordinates are missing, default coordinates assignment occurs.

After instantiating and initializing all Entities of the description, the PROCESSES section is used to create the behavioral domain of the model, specifying relations between Entities, as well as behavioral aspects. A low-level behavioral rule is a NWNs transition, which can be defined as follows:

```
PROCESS(keyword:entities_declaration,
                keyword:entities_declaration,...,
                {transition function}, delay(N))
```

The `keyword:entities_declaration` pairs are used to specify the input/output properties of the transition: the `keyword`, which can be `IN`, OUT, or `BI`, specifying the direction of the arcs from and to the entities listed in the `entities_declaration` as:

```
{entity name[arcs_number],...,
entity name[arcs_number]}
```

Incoming (`IN`) arcs are responsible for consuming tokens from places, whereas outgoing arcs (OUT) are responsible for producing tokens into a place; bidirectional arcs (`BI`) are used to check the marking of a place.

The `transition function` parameter specifies enabling and firing rules in transitions:

```
{if (enabling_function) firing_function}
```

Basic operations include modification of token values and assignment of tokens to places after evaluating enabling rules. These operations can function with a partly stochastic behavior as well.

PROCESSES also have the `delay(N)` attribute, defining the amount of time elapsing between transition enabling and firing. This attribute is specified using the time unit from the `INIT` section.

CHANNELS use a particular transition firing function. In fact, they necessarily interlock transitions with at least another Module. This sets up a synchronous communcation mechanism between Modules, having a direction, that is, the two ends of the CHANNEL are not equivalent. On one end, the *down-link* transition starts the communication, on the other hand the *up-link* transition receives the communication in a synchronous way. The *down-link* belongs to the Module carrying the CHANNEL declaration in the `ENTITIES` section, while the *up-link* engages in the communication following this syntax:

```
module_downlink:channel()
```

Token flow can proceed both from *down-link* to *up-link* and *viceversa.*

The PROCESS construct specifies which tokens are exchanged within the CHANNEL, and to which input and output places it connects, with this syntax:

```
when( module downlink:channel(tk)) {
        tk = place IN.token
}
```

or

```
when( module downlink:channel(tk)) {
        place OUT.token = tk
}
```

The PROCESSES section specifies relations between Modules loaded from the librariesas the PACKAGE section describes. In this case, it takes as arguments the Entities that need to enter in relation.

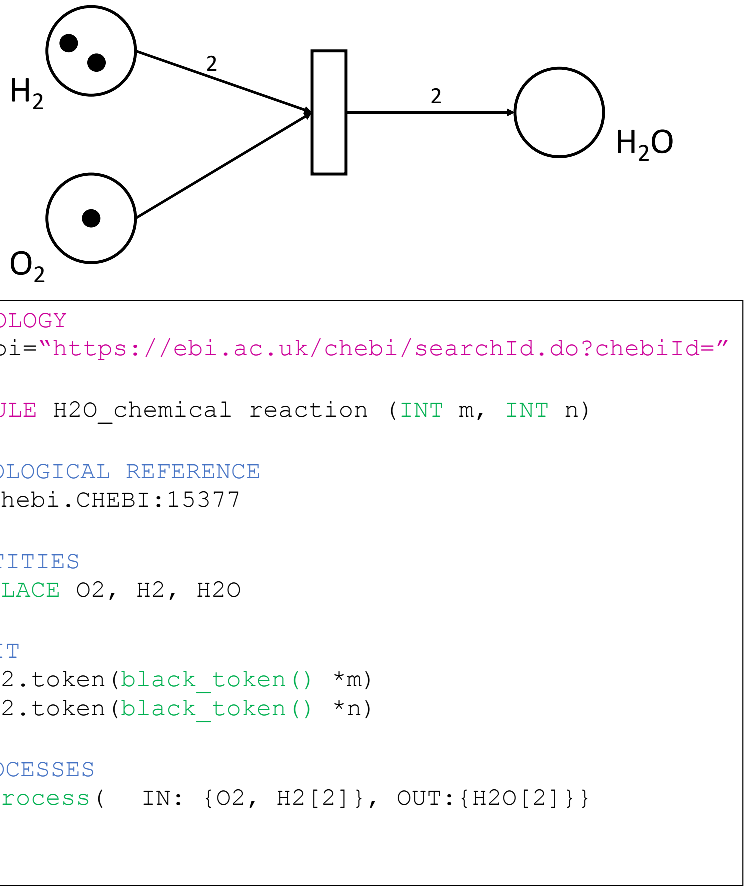
5.7 Application examples

This Section provides three application examples. The first one is the simplest one, and it aims to illustrate BiSDL syntax at the low-level, including NWNs elements manipulation. The second example aims to show the detailed BiSDL description for a model of a relatively simple biological process. The third one relates to the example provided for the NWNs modeling approach to ontogenesis and primarily aims at giving a hint about user experiences, describing both the expert and non-expert usage modes.

A water-based toy example

To clarify the actual usage of BiSDL, we present a simple toy example involving the creation of a molecule of water. This example does not show all BiSDL potential in responding to the complex requirements posed by systems biology to representational means. It is relevant to the life sciences application domain in general, both for chemical and biochemical reactions belong to one of the relevant organization levels from complex biological systems. The choice of the molecule is appropriate also for life (as we know it) is water-based. This example intends to be simple, aiming mainly at explaining basic BiSDL functionalities.

Figure 5.6 shows this simple toy model of a chemical reaction involving H_2O. To obtain two tokens of water it is necessary to have one O_2 token and 2 H_2 tokens. The two source molecules H_2 and O_2, and the final molecule H_2O are defined as places in the `ENTITIES` section. `H2` and `O2` are initialized with *m* and *n* black tokens each in the `INIT` section. A transition from the `PROCESSES` section models the chemical reaction. In the definition of the two input arcs `O2` and `H2` the code specifies that two `H2` tokens and one `O2` are needed to enable the transition and generate two `H2O` tokens.



```
ONTOLOGY
Chebi="https://ebi.ac.uk/chebi/searchId.do?chebiId="

MODULE H2O_chemical reaction (INT m, INT n)

 BIOLOGICAL REFERENCE
   Chebi.CHEBI:15377

 ENTITIES
   PLACE O2, H2, H2O

 INIT
   O2.token(black_token() *m)
   H2.token(black_token() *n)

 PROCESSES
   process(  IN: {O2, H2[2]}, OUT:{H2O[2]}}

END
```

Figure 5.6. H_2O chemical reaction. Taken from Muggianu et al., 2018.

This Module models the chemical reaction directly manipulating elements from the Petri Nets formalism. Once created, the Module belongs in the `ENTITIES` section of another Module, for example as:

```
ENTITIES
  ENTITY H2O_react
  ....
INIT
  ....
  H2O_react = H2O_chemical_reaction(1,2)
```

In the following Section, additional application examples focus on other BiSDL capabilities. In particular, it presents the descriptions of a signaling cascade, and that of the application example from Section 4.4.2, illustrating the modeling approach targeting ontogenesis: the VPC specification process in C. Elegans.

5.7.1 Application example: the MAPK signaling pathway

In this Section, we will present an example of BiSDL modeling of the *RAS/RAF/MAPK* signaling pathway shown in Figure 5.7. This pathway is part of more extensive model description, presented in the following Section.

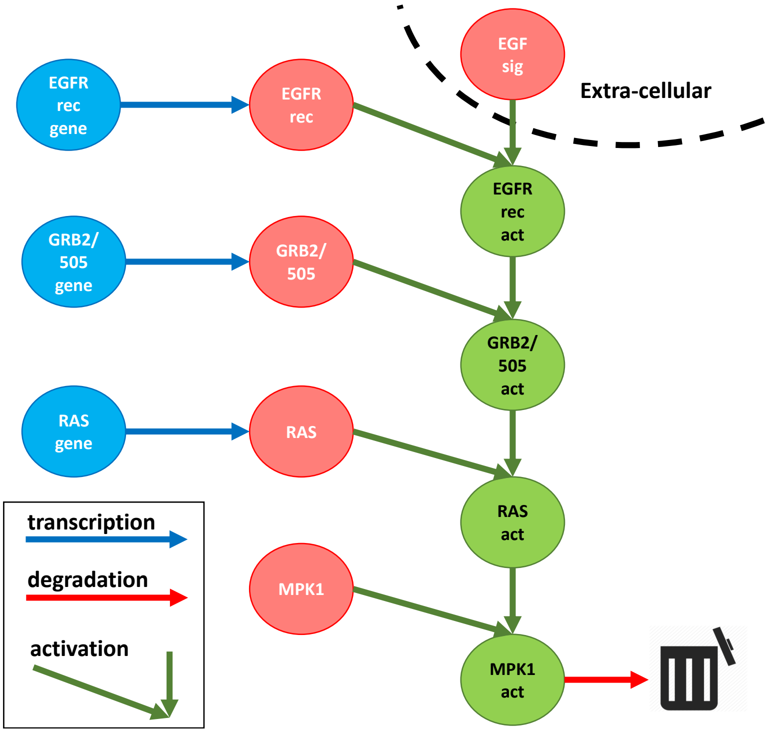


Figure 5.7. *RAS/RAF/MAPK* signalling pathway. Taken from Muggianu et al., 2018.

The extracellular *EGF-like* signal binds with its receptor (transcribed from the *EGFR-like* receptor gene): this is where the pathway starts. The *EGFR-like* receptor activates in turn, and binds and activates the *Grb2/Sos* protein. This protein reacts with and activates the *RAS* protein, which finally binds and activates the *MPK1* protein generating the final *MPK1* activated protein. This pathway ends in the production of the *Map Kinase Kinase* active protein, the only protein in the model for which degradation works explicitly.

The BiSDL description of the pathway includes Transcription, Activation, and Degradation building blocks. In Figure 5.8, the general template presented in 5.6.1 shows the top-level BiSDL description of the model.

First of all, the code includes the `PathwayRegulations` library, comprising the necessary building blocks, and links to the ontologies that will be employed.

In the `MODULE` declaration the code defines the model interface, which includes: an *EGF-like* signal, the *MPK1* protein and its activated form, and an *EGFR-like* receptor. All the entities required to model the pathway are listed: the *RAS* and *Grb2/Sos* proteins (with their activated forms), and a *EGFR-like* activated receptor protein.

The `INIT` section provides links to the employed existing Modules. In this case, all of them link to a `simple_protein model`. As shown in Figure 5.9, the `simple_protein` is modeled as a NWNs Place. Required parameters are the name of the protein, an ID from one of the included ontologies, and a token type. In this example, the `INIT` section creates a Place for each protein and imposes a marking to each place.

```
PACKAGE PathwaysRegulations
ONTOLOGY Proteins =
″https://research.bioinformatics.udel.edu/pro/entry/PR%3A″
ONTOLOGY Gene      =
″https://www.ncbi.nlm.nih.gov/gene?cmd=Retrieve&dopt=full_
report&list_uids=″
ONTOLOGY PathwayOntology   =
“http://bioportal.bioontology.org/ontologies/PTS/?p=classes
&conceptid=http%3A%2F%2Fscai.fraunhofer.de%2FPWDICT%23″

MODULE RAS_RAF_MAPK_sig_path (ENTITY EGF-like_signal,
                              ENTITY mpk_1_act,
                              ENTITY mpk_1,
                              ENTITY EGFR-likeRec)

  BIOLOGICAL REFERENCE
    PathwayOntology.ID0176

  ENTITIES
    ENTITY RAS, RAS_act, EGFR-likeRec_act
    ENTITY GRB-2/505, GRB-2/505_act

  INIT
    GRB-2/505 = simple_protein(“GRB2/505″,
                               Proteins.000008220,
                               black_token()*3)
    RAS = simple_protein(“RAS″,Proteins.000013743,
                         black_token()*3)
    GRB-2/505_act = simple_protein(“GRB-2/505_act″,
                    Proteins.000008220,black_token()*3)
     RAS_act = simple_protein(“RAS_act″,
                 Proteins.000013743,black_token()*3)
     EGFR-likeRec_act = simple_protein(“EGFR-likeRec_act″,
                 Proteins.000006933,black_token()*3)

  PROCESSES
    transcription(RAS, Gene.3845)
    transcription(EGFR-likeRec, Gene.1956)
    transcription(GRB-2/505, Gene.2885)
    activation(EGFR-likeRec, EGF-like_signal,
               EGFR-likeRec_act)
    activation(GRB-2/505, EGFR-likeRec_act,
               GRB-2/505_act )
    activation(RAS, GRB-2/505_act, RAS_act)
    activation(mpk_1, RAS_act, mpk_1_act)
    degradation(mpk_1_act, 100)
END
```

Figure 5.8. *RAS/RAF/MAPK* signalling pathway BiSDL model. Taken from Muggianu et al., 2018.

```
MODULE simple_protein(STRING name, STRING ID, TOKEN token)

  BIOLOGICAL REFERENCE
    ID
  ENTITIES
    PLACE protein
  INIT
    protein.name = name
    protein.token(token)
END
```

```
MODULE simple_gene (STRING name, TOKEN token, STRING ID)

  BIOLOGICAL REFERENCE
    ID
  ENTITIES
    PLACE gene
  INIT
    gene.name = name
    gene.token( token )
END
```

```
MODULE transcription (ENTITY protein, STRING ID)

  ENTITIES
    PLACE gene
  INIT
    gene = simple gene(protein.name + ″_wt″,ID,
                       black_token())
  PROCESSES
    process(BI:{gene}, OUT:{protein} )

END
```

```
MODULE degradation (ENTITY protein, INT n)

  PROCESSES
    process(IN:{protein}, delay(n) )
END
```

```
MODULE activation (ENTITY protein_to_activate,
                   ENTITY protein_active,
                   ENTITY protein_act)

  PROCESSES
    process(BI:{protein_active[4],protein_to_activate[4]},
            IN:{protein_to_activate}, OUT:{protein_act} )

END
```

Figure 5.9. BiSDL library elements. Taken from Muggianu et al., 2018.

The PROCESSES section put these entities in relation: transcription processes involve *RAS*, *Grb2/Sos*, and *EGFR-likeRec*.

The Transcription Module creates a new place modeled by the simple_gene module. Therefore, with the three transcription processes, the model includes three new places, each initialized with one black token, modeling the genes that transcribe the required proteins. The PROCESSES section of the transcription Module contains a transition operating the creation of tokens from ”gene” to ”protein” places.

Activation Modules model the passage of a protein to its active state. They require three Entities: the inactive protein (protein_to_activate), the protein (protein_active) performing activtion, and the active protein (protein_act). The PROCESSES section of the Activation Module has a transition creating a token in the protein_act place when enough tokens are present in the protein_to_activate and protein_active places.

Degradation Modules describe mpk_1_act degradation, having a transition which consumes tokens from the ”input” protein place after a time delay.

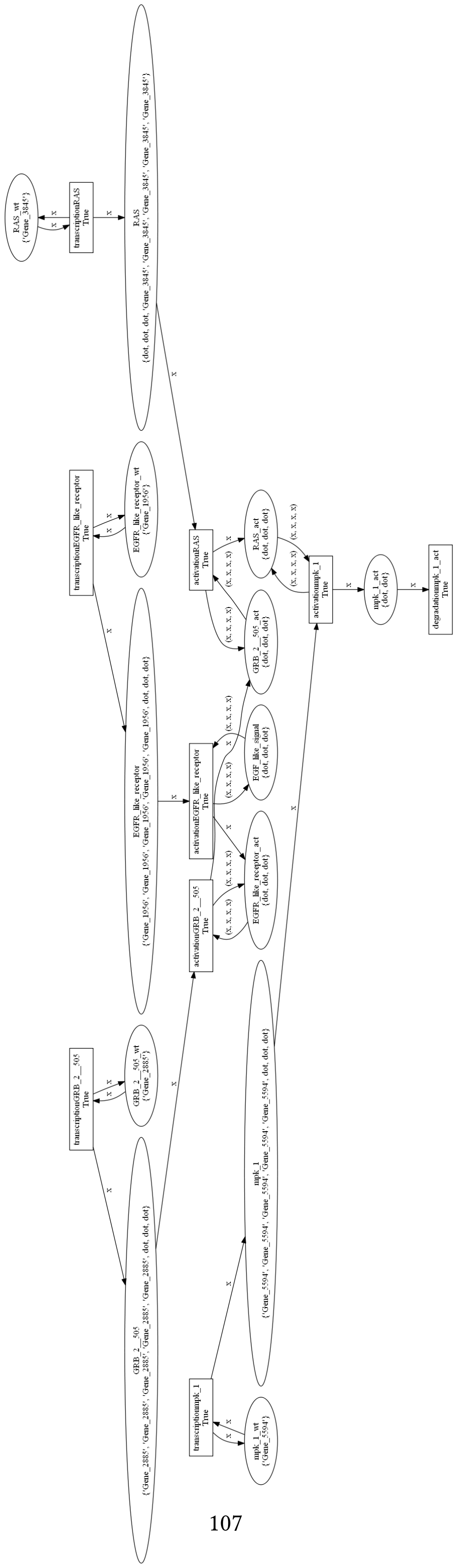


Figure 5.10. Visual representation of the model in SNAKES generated with the GraphViz plugin. Taken from Muggianu et al., 2018.

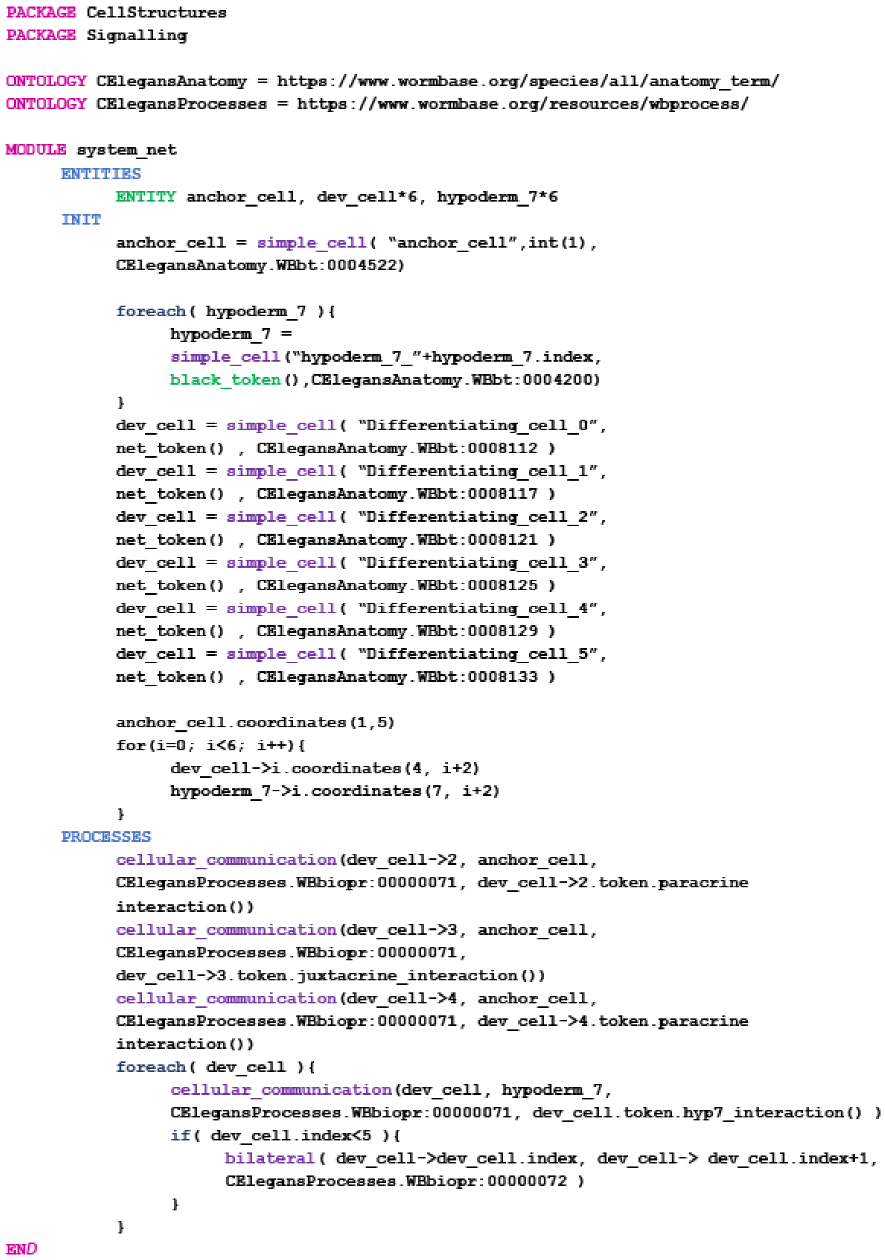
5.7.2 Application example: VPC specification in C. Elegans

Section 4.4.2 recapitulates into details the biological process of VPC specification in C. Elegans, presenting its NWNs model and its simulation. A BiSDL description of that model completes the example, providing an overview of the presented framework. The main objective of this Section is to provide the reader with a hint about the user experience of BiSDL, from both the expert and non-expert perspectives. For a full implementation of the BiSDL code, see Flavia Muggianu, 2018. The complex Patterning Functional Module and several Building Blocks compose the VPC Specification NWNs model. This Section recapitulates how different users can approach BiSDL dealing with this example.

The expert user encodes knowledge By building up Modules from scratch, or modifying existing ones, expert users manipulate the BiSDL description at every abstraction level. In particular, in the presented examples existing Modules for the Cell Model are assembled to form a complete description.

This section aims to provide a flavor of how a BiSDL-based model construction process can work. A first step, in this case, is to describe using BiSDL the building blocks from the VPC modeling example, and drawing the proper interconnections and hierarchical relations between Modules.

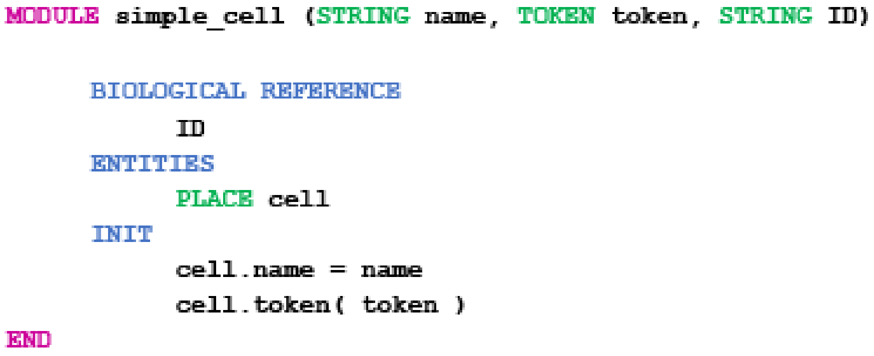
The user provides a first description of the top layer in the model, devising ISG and DL elements. Figure 5.11 shows the description corresponding to the ISG and DL model in section 4.4.2.



```
PACKAGE CellStructures
PACKAGE Signalling

ONTOLOGY CElegansAnatomy = https://www.wormbase.org/species/all/anatomy_term/
ONTOLOGY CElegansProcesses = https://www.wormbase.org/resources/wbprocess/

MODULE system_net
    ENTITIES
        ENTITY anchor_cell, dev_cell*6, hypoderm_7*6
    INIT
        anchor_cell = simple_cell( “anchor_cell”,int(1),
        CElegansAnatomy.WBbt:0004522)

        foreach( hypoderm_7 ){
            hypoderm_7 =
            simple_cell(“hypoderm_7_”+hypoderm_7.index,
            black_token(),CElegansAnatomy.WBbt:0004200)
        }
        dev_cell = simple_cell( “Differentiating_cell_0”,
        net_token() , CElegansAnatomy.WBbt:0008112 )
        dev_cell = simple_cell( “Differentiating_cell_1”,
        net_token() , CElegansAnatomy.WBbt:0008117 )
        dev_cell = simple_cell( “Differentiating_cell_2”,
        net_token() , CElegansAnatomy.WBbt:0008121 )
        dev_cell = simple_cell( “Differentiating_cell_3”,
        net_token() , CElegansAnatomy.WBbt:0008125 )
        dev_cell = simple_cell( “Differentiating_cell_4”,
        net_token() , CElegansAnatomy.WBbt:0008129 )
        dev_cell = simple_cell( “Differentiating_cell_5”,
        net_token() , CElegansAnatomy.WBbt:0008133 )

        anchor_cell.coordinates(1,5)
        for(i=0; i<6; i++){
            dev_cell->i.coordinates(4, i+2)
            hypoderm_7->i.coordinates(7, i+2)
        }
    PROCESSES
        cellular_communication(dev_cell->2, anchor_cell,
        CElegansProcesses.WBbiopr:00000071, dev_cell->2.token.paracrine
        interaction())
        cellular_communication(dev_cell->3, anchor_cell,
        CElegansProcesses.WBbiopr:00000071,
        dev_cell->3.token.juxtacrine_interaction())
        cellular_communication(dev_cell->4, anchor_cell,
        CElegansProcesses.WBbiopr:00000071, dev_cell->4.token.paracrine
        interaction())
        foreach( dev_cell ){
            cellular_communication(dev_cell, hypoderm_7,
            CElegansProcesses.WBbiopr:00000071, dev_cell.token.hyp7_interaction() )
            if( dev_cell.index<5 ){
                bilateral( dev_cell->dev_cell.index, dev_cell-> dev_cell.index+1,
                CElegansProcesses.WBbiopr:00000072 )
            }
        }
END
```

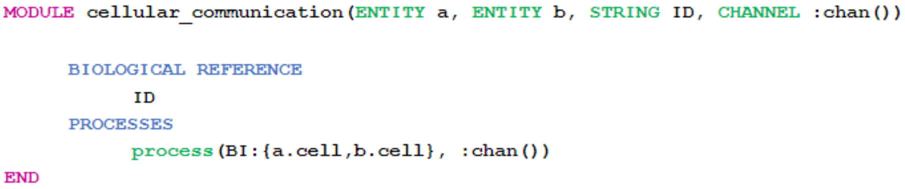
Figure 5.11. BiSDL Module for the ISG and DL models for VPC specification. Taken from Muggianu et al., 2018.

After this, each specific Module from the loaded PACKAGES needs to be specified. For the CellStructures PACKAGE, simple cell (Figure 5.12), cellular communication (Figure 5.13) and bilateral (Figure 5.14) cover the interactions taking place between cell positions in the ISG.



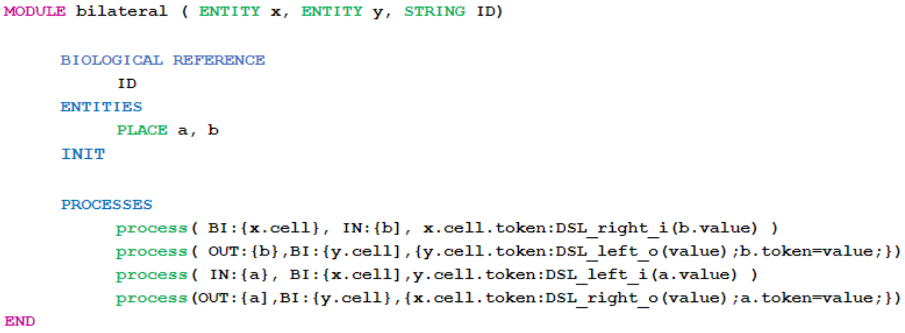
```
MODULE simple_cell (STRING name, TOKEN token, STRING ID)

        BIOLOGICAL REFERENCE
                ID
        ENTITIES
                PLACE cell
        INIT
                cell.name = name
                cell.token( token )
END
```

Figure 5.12. BiSDL description for the simple cell Module. Taken from Muggianu et al., 2018.



```
MODULE cellular_communication(ENTITY a, ENTITY b, STRING ID, CHANNEL :chan())

        BIOLOGICAL REFERENCE
              ID
        PROCESSES
              process(BI:{a.cell,b.cell}, :chan())
END
```

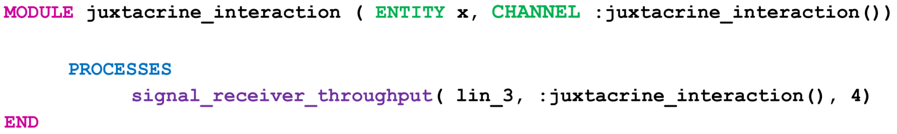
Figure 5.13. BiSDL description for the cellular communication Module. Taken from Muggianu et al., 2018.



```
MODULE bilateral ( ENTITY x, ENTITY y, STRING ID)

        BIOLOGICAL REFERENCE
              ID
        ENTITIES
              PLACE a, b
        INIT

        PROCESSES
              process( BI:{x.cell}, IN:{b], x.cell.token:DSL_right_i(b.value) )
              process( OUT:{b},BI:{y.cell],{y.cell.token:DSL_left_o(value);b.token=value;})
              process( IN:{a}, BI:{x.cell],y.cell.token:DSL_left_i(a.value) )
              process(OUT:{a],BI:{y.cell},{x.cell.token:DSL_right_o(value);a.token=value;})
END
```

Figure 5.14. BiSDL description for the bilateral Module. Taken from Muggianu et al., 2018.

The `Signalling` PACKAGE relies on the juxtacrine interaction (Figure 5.15), paracrine interaction (Figure 5.16) and hyp7 interaction (Figure 5.17) Modules.



```
MODULE juxtacrine_interaction ( ENTITY x, CHANNEL :juxtacrine_interaction())

        PROCESSES
              signal_receiver_throughput( lin_3, :juxtacrine_interaction(), 4)
END
```

Figure 5.15. BiSDL description for the juxtacrine interaction Module. Taken from Muggianu et al., 2018.

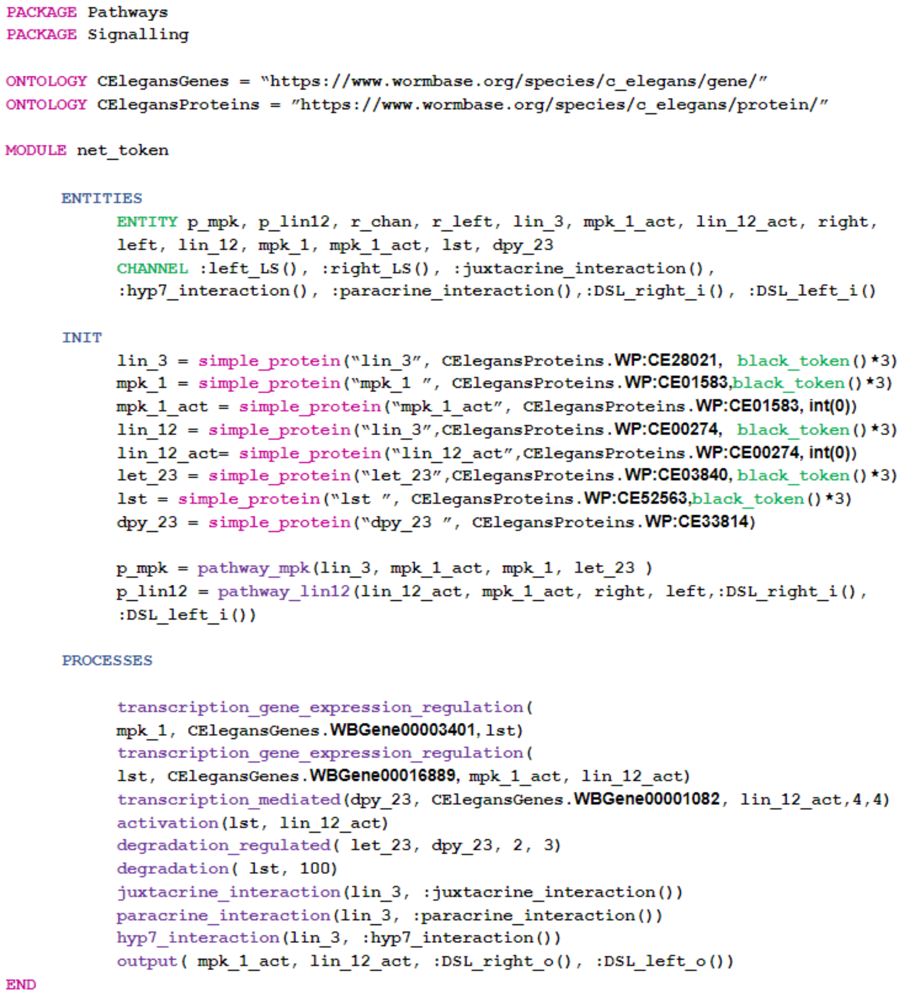
```
MODULE paracrine_interaction ( ENTITY x, CHANNEL :paracrine_interaction())

     PROCESSES
          signal_receiver_throughput( lin_3, :paracrine_interaction(), 2)

END
```

Figure 5.16. BiSDL description for the paracrine interaction Module. Taken from Muggianu et al., 2018.

```
MODULE hyp_7_interaction ( ENTITY lin_3, CHANNEL :hyp7_interaction() )

     PROCESSES
          signal_receiver_throughput( lin_3, :hyp7_interaction(), 1)
END
```

Figure 5.17. BiSDL description for the hyp7 interaction Module. Taken from Muggianu et al., 2018.

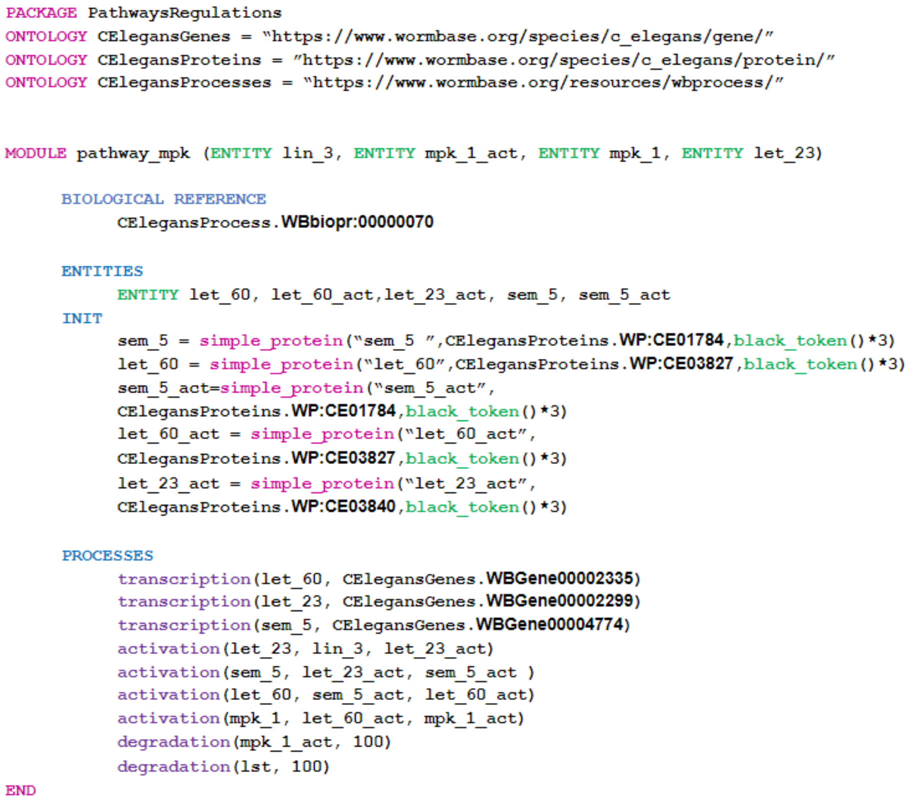
After that, the user describes Modules relative to the model of Pn.p cells. Figure 5.18 shows the description corresponding to the Cell model in section 4.4.2.



```
PACKAGE Pathways
PACKAGE Signalling

ONTOLOGY CElegansGenes = “https://www.wormbase.org/species/c_elegans/gene/”
ONTOLOGY CElegansProteins = ”https://www.wormbase.org/species/c_elegans/protein/”

MODULE net_token

     ENTITIES
          ENTITY p_mpk, p_lin12, r_chan, r_left, lin_3, mpk_1_act, lin_12_act, right,
          left, lin_12, mpk_1, mpk_1_act, lst, dpy_23
          CHANNEL :left_LS(), :right_LS(), :juxtacrine_interaction(),
          :hyp7_interaction(), :paracrine_interaction(),:DSL_right_i(), :DSL_left_i()

     INIT
          lin_3 = simple_protein(“lin_3”, CElegansProteins.WP:CE28021, black_token()*3)
          mpk_1 = simple_protein(“mpk_1 ”, CElegansProteins.WP:CE01583,black_token()*3)
          mpk_1_act = simple_protein(“mpk_1_act”, CElegansProteins.WP:CE01583, int(0))
          lin_12 = simple_protein(“lin_3”,CElegansProteins.WP:CE00274, black_token()*3)
          lin_12_act= simple_protein(“lin_12_act”,CElegansProteins.WP:CE00274, int(0))
          let_23 = simple_protein(“let_23”,CElegansProteins.WP:CE03840,black_token()*3)
          lst = simple_protein(“lst ”, CElegansProteins.WP:CE52563,black_token()*3)
          dpy_23 = simple_protein(“dpy_23 ”, CElegansProteins.WP:CE33814)

          p_mpk = pathway_mpk(lin_3, mpk_1_act, mpk_1, let_23 )
          p_lin12 = pathway_lin12(lin_12_act, mpk_1_act, right, left,:DSL_right_i(),
          :DSL_left_i())

     PROCESSES

          transcription_gene_expression_regulation(
          mpk_1, CElegansGenes.WBGene00003401, lst)
          transcription_gene_expression_regulation(
          lst, CElegansGenes.WBGene00016889, mpk_1_act, lin_12_act)
          transcription_mediated(dpy_23, CElegansGenes.WBGene00001082, lin_12_act,4,4)
          activation(lst, lin_12_act)
          degradation_regulated( let_23, dpy_23, 2, 3)
          degradation( lst, 100)
          juxtacrine_interaction(lin_3, :juxtacrine_interaction())
          paracrine_interaction(lin_3, :paracrine_interaction())
          hyp7_interaction(lin_3, :hyp7_interaction())
          output( mpk_1_act, lin_12_act, :DSL_right_o(), :DSL_left_o())
END
```

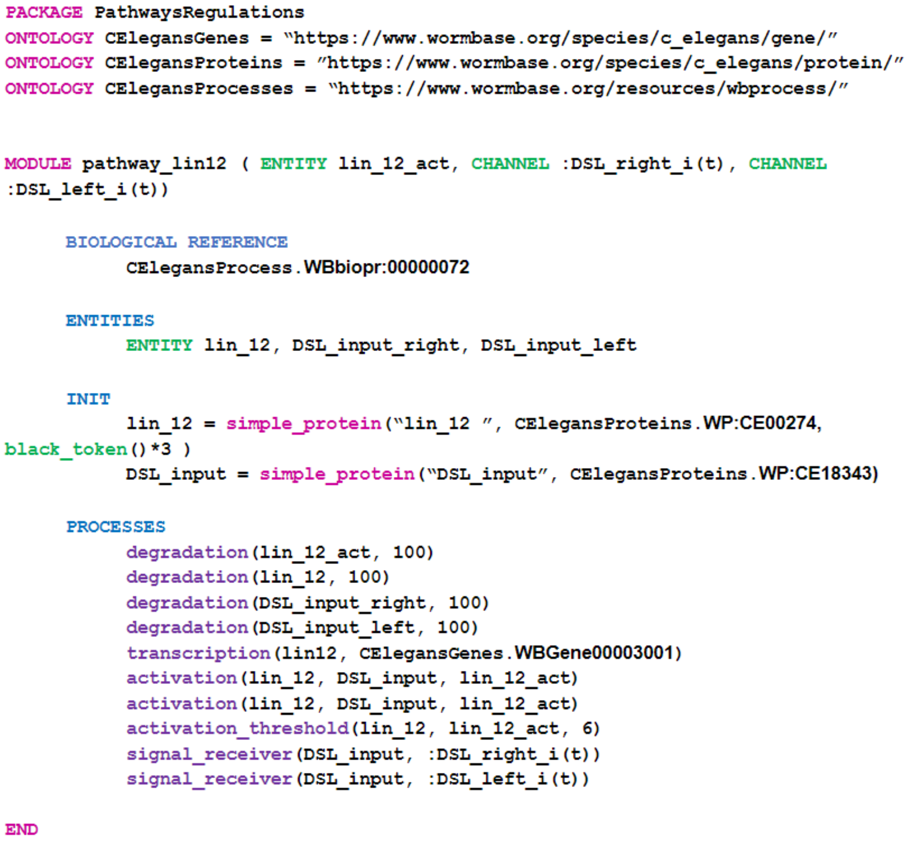
Figure 5.18. BiSDL Module for the Cell model in the VPC specification example. Taken from Muggianu et al., 2018.

After this, each specific Module from the loaded PACKAGES is specified. For the Pathways PACKAGE, the LIN3/LET23 (Figure 5.19) and the DSL/LIN12 (Figure 5.20) pathways are described.



```
PACKAGE PathwaysRegulations
ONTOLOGY CElegansGenes = “https://www.wormbase.org/species/c_elegans/gene/”
ONTOLOGY CElegansProteins = ”https://www.wormbase.org/species/c_elegans/protein/”
ONTOLOGY CElegansProcesses = “https://www.wormbase.org/resources/wbprocess/”


MODULE pathway_mpk (ENTITY lin_3, ENTITY mpk_1_act, ENTITY mpk_1, ENTITY let_23)

      BIOLOGICAL REFERENCE
            CElegansProcess.WBbiopr:00000070

      ENTITIES
            ENTITY let_60, let_60_act,let_23_act, sem_5, sem_5_act
      INIT
            sem_5 = simple_protein(“sem_5 ”,CElegansProteins.WP:CE01784,black_token()*3)
            let_60 = simple_protein(“let_60”,CElegansProteins.WP:CE03827,black_token()*3)
            sem_5_act=simple_protein(“sem_5_act”,
            CElegansProteins.WP:CE01784,black_token()*3)
            let_60_act = simple_protein(“let_60_act”,
            CElegansProteins.WP:CE03827,black_token()*3)
            let_23_act = simple_protein(“let_23_act”,
            CElegansProteins.WP:CE03840,black_token()*3)

      PROCESSES
            transcription(let_60, CElegansGenes.WBGene00002335)
            transcription(let_23, CElegansGenes.WBGene00002299)
            transcription(sem_5, CElegansGenes.WBGene00004774)
            activation(let_23, lin_3, let_23_act)
            activation(sem_5, let_23_act, sem_5_act )
            activation(let_60, sem_5_act, let_60_act)
            activation(mpk_1, let_60_act, mpk_1_act)
            degradation(mpk_1_act, 100)
            degradation(lst, 100)
END
```

Figure 5.19. BiSDL Module for the LIN3/LET23 pathway from the Cell model in the VPC specification example. Taken from Muggianu et al., 2018.



```
PACKAGE PathwaysRegulations
ONTOLOGY CElegansGenes = “https://www.wormbase.org/species/c_elegans/gene/”
ONTOLOGY CElegansProteins = ”https://www.wormbase.org/species/c_elegans/protein/”
ONTOLOGY CElegansProcesses = “https://www.wormbase.org/resources/wbprocess/”


MODULE pathway_lin12 ( ENTITY lin_12_act, CHANNEL :DSL_right_i(t), CHANNEL
:DSL_left_i(t))

      BIOLOGICAL REFERENCE
            CElegansProcess.WBbiopr:00000072

      ENTITIES
            ENTITY lin_12, DSL_input_right, DSL_input_left

      INIT
            lin_12 = simple_protein(“lin_12 ”, CElegansProteins.WP:CE00274,
black_token()*3 )
            DSL_input = simple_protein(“DSL_input”, CElegansProteins.WP:CE18343)

      PROCESSES
            degradation(lin_12_act, 100)
            degradation(lin_12, 100)
            degradation(DSL_input_right, 100)
            degradation(DSL_input_left, 100)
            transcription(lin12, CElegansGenes.WBGene00003001)
            activation(lin_12, DSL_input, lin_12_act)
            activation(lin_12, DSL_input, lin_12_act)
            activation_threshold(lin_12, lin_12_act, 6)
            signal_receiver(DSL_input, :DSL_right_i(t))
            signal_receiver(DSL_input, :DSL_left_i(t))

END
```

Figure 5.20. BiSDL Module for the DSL/LIN12 pathway from the Cell model in the VPC specification example. Taken from Muggianu et al., 2018.

The LIN3/LET23 pathway recapitulates the EGFR pathway example from the previous section. For a more detailed description of the Modules employed in pathway descriptions, and for the description of those from the `Signaling` Package, see Flavia Muggianu, 2018.

The presented BiSDL-based model construction process yields as a final product the BiSDL description of the overall VPC specification process. Also, it physiologically produces, as by-products, all Modules used for composing it. Both can populate libraries, facilitating the reusability of both the overall model and its subparts.

The non-expert user re-uses knowledge The non-expert user can access BiSDL libraries, and use the Modules present in there to reuse complete BiSDL descriptions or some of the Modules composing them. This kind of use can aim both at studying the underlying model in itself, or at embedding it in larger models, or even at combining some of its subparts with other existing descriptions, creating, in this way, brand new models. BiSDL Modules have parameters that users can easily fine-tune, adjusting and customizing existing descriptions to their specific scopes.

Figure 5.21 provides a scheme of the BiSDL circular pipeline supporting knowledge encoding and decoding.

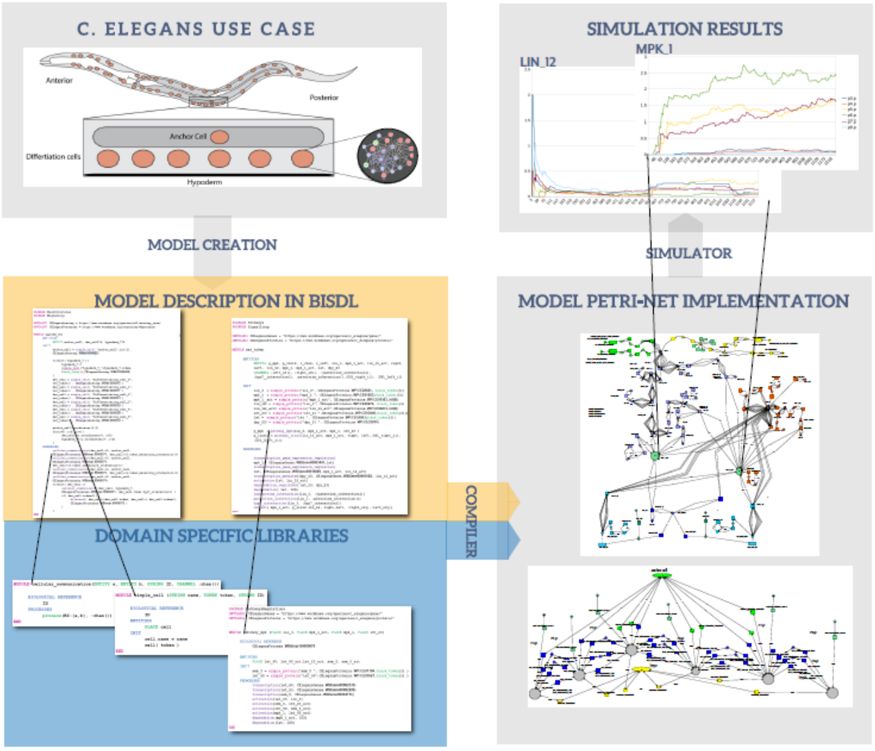


Figure 5.21. BiSDL pipeline supporting knowledge encoding and decoding for the VPC specification application example. Taken from Muggianu et al., 2018.

5.7.3 Go with the flow: from BiSDL descriptions to simulations

BiSDL descriptions are designed to automatically translated into NWNs models. A BiSDL compiler is currently under development, and a first prototypical version can translate BiSDL models into executable Python code based on a custom extension of the SNAKES Python library Pommereau, 2015, which a custom simulator can directly handle.

Figure 5.22 recapitulaes the flow from descriptions to simulations.

Figure 5.23 shows the automatically generated Python code of the `Transcription` Module, while Figure 5.10 depicts the NWNs model automatically compiled from this BiSDL description.

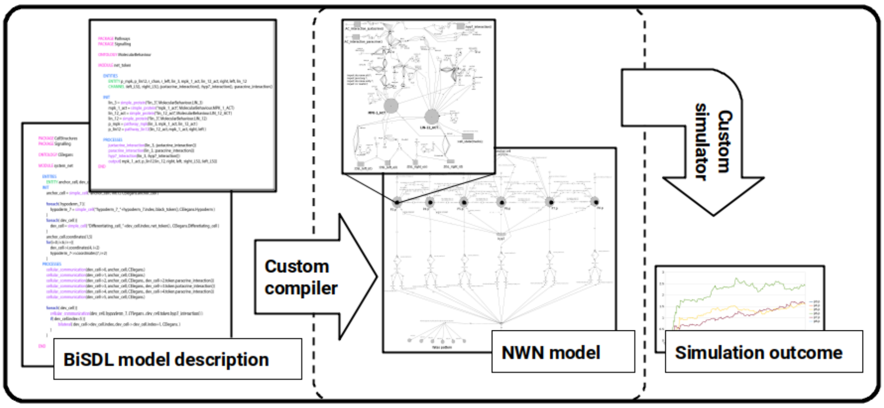


Figure 5.22. Scheme of the flow from BiSDL descriptions, to NWN models and their simulation.

```
class transcription:
    def __init__(self ,protein,ID,net):
                gene = simple_gene(protein.module_place.name + "_wt",
                   ID, [BlackToken()], net)

        process_func0=ProcessFunction()

        net.process("transcription", process_func0,
          output_places = [protein], bidir_places=[gene])
```

Figure 5.23. Phyton code snippet for transcription process. Taken from Muggianu et al., 2018.

5.8 Conclusion

BiSDL is an ongoing effort to create a new language able to model complex biological use cases. BiSDL is intended to overcome several weaknesses of existent languages and to group their strengths. Whereas other languages rely on the XML syntax, we chose a much more human-readable format to make the language biologist-friendly. To facilitate learning and the use of libraries and their link to different ontologies, we are working on the development of a complete toolkit including an editor, a parser, and a simulator. The simulator will provide a graphical user interface to help the user customize each simulation easily. While developing these examples, prototypical versions of a BiSDL compiler and a simulator for extended SNAKES NWNs models took shape. The complex of the NWNs modeling approach, the BiSDL specification, the custom compiler, and simulator constitute a prototype of the entire flow from a BiSDL description to the simulation of compiled NWNs models.

Chapter 6

Discussion

6.1 Short summary

The proposed computational framework devises a modeling approach that collects contributions from the different subdomains involved, and a high-level model description language making models accessible for the non-expert users in the field. The goal is to foster true interdisciplinarity in systems biology by creating a common playground for all the stakeholders. The resulting genuinely shared perspective should allow to ask new questions, and orient the growing technological capabilities both on the computational and high-throughput analysis techniques fronts. Ultimately, the proposed framework wants to contribute, as an enabler, to the cultural shift from multi-disciplinarity to interdisciplinarity in systems biology.

6.2 Discussion

Diversity generates value Effective collaboration allows for redefining biological complexity under a comprehensive perspective, leveraging the full spectrum of representational capabilities present in the diverse world of systems biology. A better, comprehensive and shared understanding of complexity in biology can set a starting point for methodological and instrumental paradigm shifts, in fact creating a discipline out of many. This section recapitulates the presented contributions under this perspective, pointing out the scope of the proposed framework within this scheme.

Systems biology, having entire biological systems as objects of investigations, collects the contributions of the different traditional research domains centering over each separated system portion. This approach poses the challenge to deal with diversity: different scientific profiles contribute to each subdomain, which has its particular subculture and technical language. Comprehension of diversity reflects not only in vocabulary but primarily on the cultural views emerging in representational endeavors over biological complexity. Several definitions of biological complexity exist, each one borrowed from a pre-existing concept from another domain.

Nowadays, it is reasonable to consider computational tools as that central to the quest of understanding life, that there is no more such research field as purely experimental biology. All scientific endeavors in life sciences must be considered at least in part as aided by computational tools and approaches Markowetz, 2017. For this reason, talking about diversity in systems biology it is necessary to cover not only the different sub-domains of experimental disciplines, for example, molecular biology, physiology, and genetics. Indeed, computational sciences as well find a place in these considerations, bringing to the melting pot even more different perspectives and approaches.

In the presented work, cultural diversity within systems biology appears like an excellent opportunity for redefining biological complexity by leveraging the diversity of different perspectives and representational means available.

This cultural challenge has both human and technological sides. Scientists from the different subdomains of systems biology are indeed humans, with their technical and representational capabilities, or scientific "natural" language. The approaches and tools they use for generating and managing scientific knowledge express their cultural specificities, both in macroscopic approaches and in technicalities.

In other words, systems biology needs to pass from multi-disciplinarity it exhibits in the current moment to real inter-disciplinarity. As highlighted in the Editorial titled ”How to avoid glib interdisciplinarity,” from Nature 552, 148 from 2017, very diverse scientific endeavors take with themselves the risk to fail at cross-domain communication. The risk is that all collaborative efforts result in parallel, domain-specific advancements diverging each one towards the scopes of its original field. Multi-disciplinary domains risk failing at evaluating the relevance of contributions, due to the absence of shared criteria across disciplines. Also, difficulties in communicating across boundaries pose a risk to lose the opportunity to generate knowledge and insights by directly access existing information and interpreting it under a different perspective than that which generated it.

To effectively leverage diversity, and transform systems biology into a genuinely interdisciplinary domain, technology can work as an enabler for collaboration between scientists with different backgrounds. This transformation requires to take care of both the human and technological sides to the challenge.

A central point to this Ph.D. work is to define a role for computational tools in this context. In particular, the presented computational framework intends to support the collaboration of different researchers, providing accessible knowledge management tools making them able to speak a common language. In this context, diversity refers to the coexistence of experimentalists from different subdomains in life sciences, computational biologists, bioinformaticians, synthetic biologists and researchers from computational and information sciences. Also, it refers to researchers intending to perform bottom-up explorations of the system, and engineers having a top-down, design-oriented approach to biology.

Given this primary objective, the proposed framework also responds to specific requirements posed by biological complexity itself.

Knowledge representation and inference In particular, in the last decades, fast technological advancements in both analytical and computational techniques allowed for the production and analysis of growing bodies of data. While this sets the basis for the construction of a large and ever-growing body of black-box predictive models of biology, on the other hand, it hinders a genuine understanding of systems functioning. Also, black-box models fail to function as informative knowledge bases, and thus to support knowledge representation and exchange even between scientists from the same subdomain. Also, they do not provide any mechanistic insight over the system, failing to support any possible engineering process relying on a clear link between system structures and implementable functions. For this reason, the proposed framework comprises a modeling approach that organizes data-driven insights into existing expert knowledge representations. In this way, it leverages technological capabilities within the frame of current understanding, in order to either enrich or contradict existing hypotheses.

Suitably representing the existing body of knowledge not only serves the scopes of state-of-the-art knowledge exchange. It also furthers knowledge inference, by allowing to build up models that can test and generate hypotheses, guiding subsequent experimental designs and ultimately orient and guide the excellent capabilities of high-throughput analysis technologies.

Summing up, facing the increasing availability of data, the proposed framework intends to enable different scientific profiles to ask the right questions while investigating biological complexity in a genuinely collaborative manner. Also, it intends to store the answers into models, making them available and understandable for all the stakeholders so that they can pose further and better questions at the next iteration.

In particular, the presented framework tackles the knowledge management cycle of systems biology at different points. It comprises a modeling approach, responding to knowledge inference and representation requirements, and a domain-specific language, making the models accessible for all the stakeholders.

For better explaining how the proposed modeling approach intends to enable diversity-driven scientific collaboration, this paragraph invites the reader to an audacious, yet concrete abstraction. This thought has the only purpose of describing the premises and scope to the modeling approach presented, and it does not intend to make for a philosophical statement about the nature of physical objects, but it surely partially recapitulates some principles from ontic structural realism (for a proper discussion of this theory and its controversies, see McKenzie, 2017). After this necessary premise, let's consider the knowledge body collects and interprets under a shared (propositional) perspective different information out of single perceptions. Informational objects populate this shared structure, together with the stories involving them. Objects are static, out-of-time entities, defined only in reason of their properties, which also define the potential interactions they may engage in. Stories are sequences of events objects can take part in, with either active or passive roles, along time. Knowledge exploration, under this scheme, devises two modes: object manipulation, aimed to extract out-of-time interaction capabilities from defined structures, and narration, aimed to extract procedural information concerning different interaction schemes emerging as sequences of events organized in time.

In systems biology, knowledge comprises both objects and stories: available information concerns either biological structures, belonging to any system organization level, or biological processes, involving functional activations of such structures. Using the vocabulary from well-known biological knowledge bases, biological entities and biological processes map to objects and stories, respectively. That is, biological entities carry information about the structure and potential interactions, and biological processes recapitulate the sequence of events taking place in time if a particular subset of potential interactions for the entities involved takes place, one after the other.

Taking this notion to the task of considering different representational means, it easily relates to the difference between denotational and operational semantics. The first one better suits the representation of static structures, naturally drawing potential, out-of-time connections between sub-structures, and defining features of biological systems in reason of the functional interaction schemes they could generate. It is possible to infer knowledge from mathematical models following this semantic by analyzing their structure, besides by simulating them. Operational semantics more naturally represents sequences of events, and their main focus is to make the temporal dynamics emerge from the formal description of a system. They tend to be analyzed by exploring the states space they can generate, or in general by execution.

It is necessary that a representational tool manages both biological entities and processes naturally since they are artifacts inherent to knowledge representation of biological complexity.

In addition to that, the fact systems biology includes entities and processes from multiple system levels poses the need to represent the underlying system hierarchy, and the possible cross-level connections. It is necessary to perform an additional abstraction from the system, since many of these connections only model the links between multiple representations of the same biological entity, each one living at different levels in the model, but correspond to the same structure in the physical world.

The inherent stochastic behavior to living systems challenges the representation of biological processes. Each punctual interaction in a process can yield, with more or less known probabilities, different outcomes. In this way, each process explodes in all the possible paths it can undergo considering all the combinations of underlying stochastic choices for interacting biological entities.

Defined by the potential interactions they can engage into, biological entities can be represented, at each time along a process, by the interactions they currently engage into. For this reason, the context they currently live into has the potential to furtherly reduce the set of possible interactions for the biological entity. At the same time, a biological process can devise a context change for a biological entity, potentially yielding a shift in the scheme of potential interactions it can undergo after that. These shifts can affect the overall system behavior in reason of different events characterizing the biological process. The intention to represent this shift poses the need for an even higher-level abstraction, that is, the segregation of biological structures and the contexts they may find themselves into along a biological process.

Since contexts themselves are nothing more than the collection of potential touchpoints between the considered biological entity and other entities from the system, this reasoning holds for those other entities as well. The necessity to further the abstraction from the system emerges: representational means need to segregate the capabilities for interaction from the biological entities involved - all of them. In this way, the context is a composite biological entity itself, and it is possible to flexibly mediate the interactions between entities accurately at each time point along the overall biological process.

To better organize this movement towards the dynamic segregation of biological entities from their context, it is possible to perform a further distinction: different aspects compose the context for a system and its subparts. In choosing the aspects to consider, the proposed modeling approach goes back to the distinction between abstract objects and stories. That is, two perspectives over the overall context for each biological entity in the system find explicit representation. The first one recapitulates spatial aspects to the biological system. In each moment of the overall process, this spatial model mediates the collapse of the potential interactions scheme into a specific scheme of actual interactions. This mediation, in the physical system, relies on spatial proximity between entities to decide whether each potential interaction takes place at that moment Bardini, Politano, et al., 2017a. The second one intends to provide a process-related context to the biological entity, which, similarly to the spatial model, reduces the set of possible interactions the biological entities can undergo, basing on the premise they need to coexist in time in order to interact. In this way, it is possible to embed into representations kinetic aspects from multiple system levels and make them relevant for the overall system evolution. Moreover, more importantly, the combination of the two contextual models draw a dynamic landscape for the overall system to evolve into which is abstract from the system itself but allows its parts to interact between each other following context-based constraints.

This approach bounds the scheme of potential interactions characterizing the overall system - considered as an object - making it collapse into specific contexts, selecting the actual interactions taking place along the overall biological process - considered as a story.

In addition to that, systems biology not always provides full resolution in describing complex structures and behaviors of biological systems. Also, not all models intend to provide a full-detail representation of a system. Information availability and the modeling scope impose in concert great flexibility to representational means, which need to allow for different degrees of abstraction in the same model. Flexible abstractions also allow to limit model complexity while preserving model accuracy: models can get simpler while their most relevant parts preserve a higher level of detail. In Bulik et al., 2009, the parametrization of the model of a metabolic network generates detailed mechanistic equations for the most relevant model parts and more abstract representations for the rest of the model.

All these aspects considered together define the strategy underlying the multi-level and multi-context modeling approach proposed, which relies on a formalism supporting flexible abstractions, and both denotational and operational semantics. That is, it allows both to describe complex objects of biological knowledge, and to tell all of their (known) stories.

All of these aspects concerned knowledge representation and inference and could hold for each of the subdomains composing systems biology. As another objective, the proposed approach aims to comprise knowledge from all of the potential contributors to research in biology. For this reason, the formalism of choice also supports the consistent representation of different information in the same model, allowing for the natural inclusion of knowledge specified under the different formalisms emerging from domain-specific sub-cultures in systems biology. For supporting this, models need to be capable to embed both functional, hypothesis-driven information, similar to that traditionally shared by experimental biologists in diagrammatic form, and quantitative information flowing out of systematic analytic procedures over the system of interest.

The intention to infer knowledge out of models brings along the necessity to consider the traditional caveat going by the words of ”all models are wrong” Box, 1976. They are nothing but tools designed to get insights over a phenomenon, and many different models, even under the same approach, may show similar performance. Model selection is the task to pick the best fit for the modeling objective, and different strategies exist, basing on different evaluation criteria Kirk et al., 2013. The most straightforward strategy of choice relies on the Ockham’s razor principle, which in this context goes under the definition of the principle of parsimony: the simplest, or least complex model among same-performance models is the top choice Raykov and Marcoulides, 1999; Sober, 1981.

At the aim of gaining new insights over both biological entities and processes, models undergo execution. This procedure brings the complexity of the system, as represented under the presented approach, to a computational perspective. Complexity, in the computational domain, finds a measure in the number of effective parameters in the model Spiegelhalter et al., 2002. In computational models, this translates in the number of instructions to execute to simulate the system dynamics. In general, models following the presented approach end up comprising vast schemes of potential interactions. Moreover, while context-based constraints may reduce possibilities to consider along each stage of a process on a semantic level, from the syntactic, and then the computational point of view, instead, they generate additional instructions to execute to operate the model. A hierarchical structure and the communication channels across its levels increase computational complexity even more. Moreover, usually, models of this kind tend to support modeling objectives devising large portions from the system of interest. On the overall, this has a cost in terms of computational complexity.

The benefit behind this cost is the maximization of accuracy, reflecting both in the capability of the models to predict system behavior accepting a certain degree of uncertainty Steyerberg et al., 2010, as well as to guarantee its reliability as a knowledge base Macklin et al., 2014; Matthews et al., 2008.

For facing the necessary computational cost, high-performance distributed computing systems can play a decisive role Holzhutter et al., 2012. For this reason, the presented modeling framework includes a custom simulator, which is set to run on these systems.

An additional and peculiar aspect to models of systems biology concerning parameter identification needs consideration. Different models can represent the same system with comparable accuracy. This trend holds for both model architecture specification and parameter identification. Considering a fixed model structure, performing parameter identification on it can take to non-unique, and separate sets of parameters. On this topic, considering the work presented in Gutenkunst et al., 2007a, models of biology exhibit what authors define as parameter sloppiness. In other words, parameter sensitivities have loose constraints. Authors claim this is a universal feature of systems biology models, actually representing the fact that different parameter sets may correctly model system behavior. The reason they hypothesize is that the system itself exhibits high robustness to parameter perturbations. These considerations highlight some criticalities around parameter uncertainty in systems biology models, hindering the supposed reliability of models as knowledge bases Matthews et al., 2008. The authors suggest that models should better work as predictive tools Daghir-Wojtkowiak et al., 2017.

In the presented approach, models aim both at performing accurate predictions over system behavior Gutenkunst et al., 2007a, and to provide a reliable and understandable knowledge base for systems biology. Parameter uncertainty and model understandability are to be considered as important issues as prediction accuracy and seeking parsimony in model construction for maximizing computational feasibility. Also, models need to support clarity and understandability Houy et al., 2012.

Each part of a model should carry verified parameter values: model re-usability Donatelli and Rizzoli, 2008 and interoperability Bichindaritz, 2006 should lead the choices of the modeler wanting to contribute to the presented framework. Also, it could be the first step to enable the collaboration between experimentalists and theoreticians, scientists and engineers working in systems biology Macklin et al., 2014; Waltemath, Karr, Bergmann, Chelliah, Hucka, Krantz, Liebermeister, Mendes, C. J. Myers, Pir, et al., 2016a,b.

Knowledge representation and exchange Model re-usability and interoperability support knowledge exchange, but in most cases, they do only among modelers, that is, researchers who can understand and produce knowledge expressed under the modeling formalism. The modeling approach alone causes an improvement compared to the pre-existing situation since at least computational biologists from different subdomains can speak the same language when using it. But, it still limits knowledge exchange to the scopes of representation and a restricted user-base: at this point, models fail to make biological knowledge accessible for non-expert users, such as experimental biologists, and exchangeable in general.

To fully realize the objective of creating value out of the diverse range of profiles contributing to systems biology, the presented framework comprises BiSDL, a domain-specific model description language, providing an intermediary layer making model-based knowledge accessible for non-experts. Also, it is possible to automatically generate simulable models from BiSDL descriptions, allowing them to access both objects and stories from systems biology body of knowledge.

This intermediary layer has a high-level syntax following semantics which intends to be close to the natural, domain-specific language of experimental life scientists. At the same time, it is possible to automatically generate models based on the formalism supporting the presented modeling approach from BiSDL descriptions. Also, low-level BiSDL syntax supports the direct usage of the bare formalism.

In this way, expert users can build up models using low-level BiSDL, and organizing them into modular objects recapitulating the biological system of interest. Such Modules populate libraries, which the non-expert users can rely on for drawing, customizing and combining objects which are relevant to their research scopes.

Under this perspective, expert users encode knowledge into BiSDL libraries, while non-expert users decode knowledge from them. Both these figures have an active role in enriching the knowledge base with their expertise into this scheme. Encoders actively perform model construction processes, embedding available information from the state-of-the-art, while decoders, to understand, guide or improve experimental designs, actively choose how to use available modules, performing, in fact, a model composition and tuning process.

Towards a systems biology culture This scheme of contribution intends to enable the different stakeholders to share not only data and information but knowledge as well, intended as their propositional interpretation of competence over information usage in the relevant contexts.

Also, populating and using the libraries of BiSDL Modules can work as a first step in the direction of the creation of a *lingua franca* for systems biology. The final, more ambitious objective is to make all of them fluent and aware of the pragmatics in using the language. This process may involve more or less steep learning curves for the different users: modelers need to learn biology, and experimentalists the computer science behind BiSDL models. Under this perspective, BiSDL intends to function also as an educational tool, progressively taking all actors involved not only beyond a linguistic barrier but towards a shared, brand new cultural playground. In this direction, the framework only intends to provide a container for the active contributions of researchers, setting the conditions for re-definitions of biological complexity to emerge from them. A good sign of the fact a systems biology pragmatics is taking shape would be observing a computer scientist and an experimental biologist telling jokes in BiSDL which are fun for both of them.

Behind this goal, the proposed framework intends to enable and foster connection among the diversity of perspectives populating systems biology, so every actor involved can get curious and proactively contribute to advancing, leveraging the technological means available, the frontier of knowledge.

In this way, diversity in systems biology becomes an opportunity to bring value to each discipline contributing to it, transforming this multi-disciplinary approach to a genuinely inter-disciplinary one, in the direction of creating a whole new discipline out of many.

Chapter 7

Conclusions and Future Perspectives

7.1 Short summary

The framework at the moment provides a prototypical version of the complete flow from BiSDL descriptions to the simulation of NWNs models. In the future, the modeling approach should be tested for scalability, considering both a broader spectrum of intracellular mechanisms and a more significant number of cells in the system. Also, the simulator should adapt to parallel computations, so to handle more computational complexity. The framework should devise complexity reduction strategies to improve computational performances. Bioinformatic pipelines should support partly data-driven models construction processes involving not only parameter identification but also model architecture. The framework should also include model analysis routines to explore models formally. A smart user interface should embed the full flow from BiSDL descriptions to simulations allowing easy model exploration and design. This interface could also rely on a visual version of BiSDL, and simulation outcomes visualization.

7.2 Conclusions and Future Perspectives

The presented work introduces a computational framework for knowledge management in systems biology. A modeling strategy and a model description language are the main pillars of the framework, and a custom compiler and simulator complete the flow from descriptions to simulations.

This section briefly recapitulates the main contributions, highlighting current limitations and potential future developments.

The choice of the NWNs formalism supports the satisfaction of all requirements posed by biological complexity and systems biology inherent diversity. Models proposed under the presented approach function, in principle, both as good knowledge bases and prediction tools. At this stage, some limitations emerge which impose improvements in the further development of the framework.

Visual complexity Large, intricate models push the limits of visual representation capabilities of Petri Nets. When the model becomes too large, this hinders its understandability. Under this perspective, the necessity emerges to include an easy-to-use, graphical user interface for model visualization and design in the framework. This GUI, to improve understandability, should support smart zooming and de-zooming functions over models, leveraging the existing modular structure inherent to BiSDL-specified models to allow users to select which modules to explode and which ones to compress when performing knowledge explorations over the model.

Computational complexity Highly detailed models do not scale well with system size and complexity, yielding to increased computational times. The design of the flow from BiSDL to simulations needs to include, for custom simulator for extended-SNAKES models, the capability to run on distributed computing platforms, so to face this computational challenge with increased computational power. Moreover, complexity reduction algorithms should be part of the framework, making models ready to be simulated in the most efficient way when their usage is mainly oriented to perform predictions. This improvement on the overall should support good scalability of the proposed approach not only over different application examples but also over increasing system complexity to represent.

Data-driven model construction The presented approach naturally supports both hypothesis-driven and data-driven information processing in building up models. It is easy to translate the diagrammatic representation of an entire scheme of relations into a graph-based model. Data-driven procedures, at this stage, only involve the identification of parameter values in the architecture. That is, their scope is to provide quantitative information over functional relationships already present in the model architecture. In the future, it would be useful to comprise in the proposed framework automatic pipelines for data-driven model construction, supporting the at least partial design of model architecture from data. A first application could target on single-cell RNA-seq data, providing a transcriptional profile of single cells, allowing to identify them, and study their heterogeneity and state changes along with different processes in pseudotime Mayer et al., 2018; Trapnell et al., 2014. The transcriptional profile of each cell could automatically generate a minimal Cell model; this Cell model could undergo network enrichment to include new omic pools than transcriptomics; pseudotime analysis could set up the Differentiative Landscape model; complementary spatial gene expression data could build up and initialize the Interactive Spatial Grid. The framework should have the capability to automatize, at least partially, a mainly data-driven model construction process.

Model analysis Models following the presented modeling approach can undergo exploration by both analysis and formal verification. Some execution-based techniques such as model checking apply to systems biology models Kwiatkowska et al., 2008. The proposed framework should include a toolkit for model analysis, to gain deeper insights and explore the knowledge stored in it in different ways. In particular, it could be interesting to explore States Landscape (or Differentiative Landscape) models in reason of the informational entropy associated to each Cell model state, in order to build up a further possibility for interpretation of the observed phenomena, and for the generation of new hypotheses about the states changes cells undergo across such landscapes.

Flow The complete flow at this stage devises a custom compiler generating NWNs models out of BiSDL descriptions, specifying them using a custom extension of the SNAKES Python library, as well as a custom simulator. The compiler and the simulator are functioning prototypes, and their separate and joint functioning needs to undergo systematic testing phases for usability and robustness. A basic prototype of the GUI is under development, and it will need further efforts for making the framework more comfortable to use. This interface should support both smart model exploration, and easy model construction, possibly providing a visual version of BiSDL and supporting a drag-and-drop interaction style. These improvements would be especially useful for the educational side of the framework.

Libraries BiSDL supports modularity, and modules can be created, customized and re-used. Users can create and populate libraries, from which to take existing modules to use. In order to support these actions, expert users need to populate libraries, encoding knowledge into modules and putting them into shareable packages. This process is foreseen as self-supervised, also thanks to the metadata attached to each module, allowing for tracking its creation and author.

Industrial applications The presented modeling approach supports the generation and optimization of operative protocols for automatic bioreactors. An Italian patent application for extending the scopes of the proposed framework towards bioengineering applications is under development. The computational method object of the patent application describes how the multi-context, multi-level model of a biological system and the bioreactor culturing it can generate optimized protocols given a desired scope for the process.

Chapter 8

Glossary

Biological system A biological system is a complex network of biological entities of interest. Biological entities belong to different organization levels, and each organization level is centered over a different dimensional scale. Biological systems define in fact different structures, from single biomolecules, to molecular networks, organs, organisms and populations of individuals.

Biological complexity A notion of complexity comprising recurrent features of biological systems: hierarchy, compartmentalization, selective communication, adaptation, stochasticity and concurrency.

Systems biology Systems biology as a research method is the study of biological systems which encompasses all their organization levels using quantitative reasoning, computational models and high-throughput experimental technologies. Systems biology defines an interdisciplinary research domain collecting the contributions of a diversity of different domains on both the experimental and computational sides: from molecular biology to clinical research, from mathematics to software engineering. Systems biology combines the reductionist and the holistic approaches in researching biological systems of interest.

Reductionist approach Reductionism in scientific research aims to provide explanations for ever smaller entities for the phenomena of interest.

Holistic approach Holism in scientific research aims to explain systems as wholes, that is, as more of the combinations of their subparts and their relations.

Bibliography

Alemani, Davide, Francesco Pappalardo, Marzio Pennisi, Santo Motta, and Vladimir Brusic (2012), “Combining cellular automata and lattice Boltzmann method to model multiscale avascular tumor growth coupled with nutrient diffusion and immune competition”, *Journal of Immunological Methods*, 376, 1-2, pp. 55-68.

An, Gary, Qi Mi, Joyeeta Dutta-Moscato, and Yoram Vodovotz (2009), “Agent-based models in translational systems biology”, *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1, 2, pp. 159-171.

Angelov, Plamen P (2013), *Evolving rule-based models: a tool for design of flexible adaptive systems*, Physica, vol. 92.

Annpureddy, Yashwanth, Che Liu, Georgios Fainekos, and Sriram Sankaranarayanan (2011), “S-taliro: A tool for temporal logic falsification for hybrid systems”, in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Springer, pp. 254-257.

Antsaklis, Panos J and Xenofon D Koutsoukos (2003), “Hybrid systems: Review and recent progress”, *Software-Enabled Control: Information Technology for Dynamical Systems*, pp. 273-298.

Antunes, Luísa C.S., Paolo Visca, and Kevin J. Towner (2014), “Acinetobacter baumannii: evolution of a global pathogen”, *Pathogens and Disease*, 71, 3, pp. 292-301, issn: 2049-632X, doi: 10.1111/2049-632X.12125, http://dx.doi.org/10.1111/2049-632X.12125.

Arenz, Stefan and Daniel N Wilson (2016), “Blast from the past: reassessing forgotten translation inhibitors, antibiotic selectivity, and resistance mechanisms to aid drug development”, *Molecular cell*, 61, 1, pp. 3-14.

Asahara, Takashi, Akira Takahashi, Norikatsu Yuki, Rumi Kaji, Takuya Takahashi, and Koji Nomoto (2016), “Protective Effect of a Synbiotic against Multidrug-Resistant Acinetobacter baumannii in a Murine Infection Model.” *Antimicrobial agents and chemotherapy*, 60 5, pp. 3041-50.

Asai, Y., H. Oka, T. Abe, M. Okita, K. i. Hagihara, T. Nomura, and H. Kitano (2011), “An Open Platform toward Large-Scale Multilevel Modeling and Simulation of Physiological Systems”, in *2011 IEEE/IPSJ International Symposium on Applications and the Internet*, pp. 250-255, doi: 10.1109/SAINT.2011.47, http://dx.doi.org/10.1109/SAINT.2011.47.

Asai, Yoshiyuki, Takeshi Abe, Hideki Oka, Masao Okita, Ken-ichi Hagihara, Samik Ghosh, Yukiko Matsuoka, Yoshihisa Kurachi, Taishin Nomura, and Hiroaki Kitano (2014), “A Versatile Platform for Multilevel Modeling of Physiological Systems: SBML-PHML Hybrid Modeling and Simulation”, *Advanced Biomedical Engineering*, 3, pp. 50-58, doi: 10.14326/abe.3.50.

Asai, Yoshiyuki, Yasuyuki Suzuki, Yoshiyuki Kido, Hideki Oka, Eric Heien, Masao Nakanishi, Takahito Urai, Kenichi Hagihara, Yoshihisa Kurachi, and Taishin Nomura (2008), “Specifications of insilicoML 1.0: a multilevel biophysical model description language”, *The Journal of Physiological Sciences*, 58, 7, pp. 447-458.

Baeten, Jos CM (2005), “A brief history of process algebra”, *Theoretical Computer Science*, 335, 2-3, pp. 131-146.

Bardini, Roberta, Stefano Di Carlo, Gianfranco Politano, and Alfredo Benso (2018), "Modeling antibiotic resistance in the microbiota using multi-level Petri Nets", *BMC systems biology*, 12, 6, p. 108.

Bardini, Roberta, Gianfranco Politano, Alfredo Benso, and Stefano Di Carlo (2017a), "Multi-level and hybrid modelling approaches for systems biology", *Computational and Structural Biotechnology Journal*.

— (2017b), "Using multi-level Petri nets models to simulate microbiota resistance to antibiotics", in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 128-133, doi: 10.1109/BIBM.2017.8217637.

— (2018), "Computational Tools for Applying Multi-level Models to Synthetic Biology", in *Synthetic Biology*, Springer, pp. 95-112.

Barnat, Jiri, Lubos Brim, Ivana Cerna, Sven Drazan, Jana Fabrikova, Jan Lanik, David Safranek, and Hongwu Ma (2009), "Biodivine: A framework for parallel analysis of biological models", *arXiv preprint arXiv:0910.0928*.

Bartocci, Ezio and Pietro Lio (2016), "Computational modeling, formal analysis, and tools for systems biology", *PLoS computational biology*, 12, 1, e1004591.

Bechtel, William (2017), "Systems Biology: Negotiating Between Holism and Reductionism", in *Philosophy of Systems Biology*, Springer, pp. 25-36.

Behrmann, Gerd, Alexandre David, Kim G. Larsen, Oliver Moller, Paul Pettersson, and Wang Yi (2001), "Uppaal - Present and Future", in *Proc. of 40th IEEE Conference on Decision and Control*, IEEE Computer Society Press.

Benque, David, Sam Bourton, Caitlin Cockerton, Byron Cook, Jasmin Fisher, Samin Ishtiaq, Nir Piterman, Alex Taylor, and Moshe Y Vardi (2012), "BMA: Visual tool for modeling and analyzing biological networks", in *International Conference on Computer Aided Verification*, Springer, pp. 686-692.

Benso, Alfredo, Paolo Cornale, Stefano Di Carlo, Gianfranco Politano, and Alessandro Savino (2013), "Reducing the complexity of complex gene coexpression networks by coupling multiweighted labeling with topological analysis", *BioMed research international*, 2013.

Benso, Alfredo, Stefano Di Carlo, and Gianfranco Politano (2011), "A cDNA microarray gene expression data classifier for clinical diagnostics based on graph theory", *ACM Transactions on Computational Biology and Bioinformatics*, pp. 577-591.

Benso, Alfredo, Stefano Di Carlo, Gianfranco Politano, Alessandro Savino, and Enrico Bucci (2014), "Alice in Bio-Land: Engineering Challenges in the World of Life Sciences", *IT Professional*, 16, 4, pp. 38-47.

Benso, Alfredo, Stefano Di Carlo, Gianfranco Politano, Alessandro Savino, and Alessandro Vasciaveo (2014), "An extended gene protein/products boolean network model including post-transcriptional regulation", *Theoretical Biology and Medical Modelling*, 11, 1, S5.

Benso, Alfredo, Stefano Di Carlo, Hafeez Ur Rehman, Gianfranco Politano, Alessandro Savino, Giovanni Squillero, Alessandro Vasciaveo, and Stefano Benedettini (2013), "Accounting for Post-Transcriptional Regulation in Boolean Networks Based Regulatory Models.", in *IWBBIO*, pp. 397-404.

Bichindaritz, Isabelle (2006), "Memoire: A framework for semantic interoperability of case-based reasoning systems in biology and medicine", *Artificial Intelligence in Medicine*, 36, 2, pp. 177-192.

BioPortal (2018), *Pathway Ontology*, http://bioportal.bioontology.org/ontologies/PW?p=classes&conceptid=root.

Bloch, Naamah and David Harel (2016), "The tumor as an organ: comprehensive spatial and temporal modeling of the tumor and its microenvironment", *BMC bioinformatics*, 17, 1, p. 317.

Bogomolov, Sergiy, Christian Schilling, Ezio Bartocci, Gregory Batt, Andreas Podelski, and Radu Grosu (2015), "Spacerover: Parameter synthesis for multiaffine systems beyond RoVerGeNe", *To appear*, p. 8.

Bois, Frederic Y. (2009), "GNU MCSim: Bayesian statistical inference for SBML-coded systems biology models", *Bioinformatics*, 25, 11, p. 1453, doi: 10.1093/bioinformatics/btp162, eprint: /oup/

backfile/Content_public/Journal/bioinformatics/25/11/10.1093/bioinformatics/btp162/2/btp162.pdf.

Bonabeau, Eric (2002), “Agent-based modeling: Methods and techniques for simulating human systems”, *Proceedings of the national academy of sciences*, 99, suppl 3, pp. 7280-7287.

Bonzanni, Nicola, K Anton Feenstra, Wan Fokkink, and Jaap Heringa (2014), “Petri nets are a biologist’s best friend”, in *International Conference on Formal Methods in Macro-Biology*, Springer, pp. 102-116.

Bonzanni, Nicola, Elzbieta Krepska, K. Anton Feenstra, Wan Fokkink, Thilo Kielmann, Henri Bal, and Jaap Heringa (2009), “Executing multicellular differentiation: quantitative predictive modelling of C.elegans vulval development”, *Bioinformatics*, 25, 16, pp. 2049-2056, doi: 10.1093/bioinformatics/btp355, eprint: /oup/backfile/content_public/journal/bioinformatics/25/16/10.1093_bioinformatics_btp355/2/btp355.pdf, http://dx.doi.org/10.1093/bioinformatics/btp355.

Box, George EP (1976), “Science and statistics”, *Journal of the American Statistical Association*, 71, 356, pp. 791-799.

Bruni, Roberto and Ugo Montanari (2017), “pi-Calculus”, in *Models of Computation*, Springer, pp. 287-305.

Brutlag, Douglas L et al. (1994), “Understanding the human genome”, *Scientific American: Introduction to Molecular Medicine*, pp. 153-168.

Bulik, Sascha, Sergio Grimbs, Carola Huthmacher, Joachim Selbig, and Hermann G. Holzhutter (2009), “Kinetic hybrid models composed of mechanistic and simplified enzymatic rate laws – a promising method for speeding up the kinetic modelling of complex metabolic networks”, *FEBS Journal*, 276, 2, pp. 410-424, issn: 1742-4658, doi: 10.1111/j.1742-4658.2008.06784.x.

Buti, Federico, Diletta Cacciagrano, Flavio Corradini, Emanuela Merelli, and Luca Tesei (2010), “BioShape: a spatial shape-based scale-independent simulation environment for biological systems”, *Procedia Computer Science*, 1, 1, pp. 827-835.

Cabac, Lawrence, Michael Duvigneau, Daniel Moldt, and Heiko Rolke (2005), “Modeling dynamic architectures using nets-within-nets”, in *International Conference on Application and Theory of Petri Nets*, Springer, pp. 148-167.

Cabac, Lawrence, Michael Haustermann, and David Mosteller (2016), “Renew 2.5–towards a comprehensive integrated development environment for Petri net-based applications”, in *International Conference on Applications and Theory of Petri Nets and Concurrency*, Springer, pp. 101-112.

Calzone, Laurence, Francois Fages, and Sylvain Soliman (2006), “BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge”, *Bioinformatics*, 22, 14, pp. 1805-1807.

Cameron, D Ewen, Caleb J Bashor, and James J Collins (2014), “A brief history of synthetic biology”, *Nature Reviews Microbiology*, 12, 5, p. 381.

Cannon, Robert C, Padraig Gleeson, Sharon Crook, Gautham Ganapathy, Boris Marin, Eugenio Piasini, and R Angus Silver (2014), “LEMS: a language for expressing complex biological models in concise and hierarchical form and its use in underpinning NeuroML 2”, *Frontiers in neuroinformatics*, 8, p. 79.

Carbonell-Ballestero, Max, Salva Duran-Nebreda, Raul Montanez, Ricard Sole, Javier Macia, and Carlos Rodriguez-Caso (2014), “A bottom-up characterization of transfer functions for synthetic biology designs: lessons from enzymology”, *Nucleic acids research*, 42, 22, pp. 14060-14069.

Carson, John S (2002), “Model verification and validation”, in *Simulation Coference, 2002. Proceedings of the Winter*, IEEE, vol. 1, pp. 52-58.

Centers for Disease Control and Prevention (2013), *Antibiotic Resistance Threats in the United States, 2013*, [Online] https://www.cdc.gov/drugresistance/threat-report-2013/index.html.

Chaouiya, Claudine, Duncan Berenguier, Sarah M Keating, Aurelien Naldi, Martijn P Van Iersel, Nicolas Rodriguez, Andreas Drager, Finja Buchel, Thomas Cokelaer, Bryan Kowal, et al. (2013), "SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools", *BMC systems biology*, 7, 1, p. 135.

Checa, Sara and Patrick J Prendergast (2010), "Effect of cell seeding and mechanical loading on vascularization and tissue formation inside a scaffold: a mechano-biological model using a lattice approach to simulate cell activity", *Journal of biomechanics*, 43, 5, pp. 961-968.

Christensen, Soren and Niels Damgaard Hansen (1994), "Coloured Petri nets extended with channels for synchronous communication", in *International Conference on Application and Theory of Petri Nets*, Springer, pp. 159-178.

Ciocchetta, Federica and Jane Hillston (2009), "Bio-PEPA: A framework for the modelling and analysis of biological systems", *Theoretical Computer Science*, 410, 33-34, pp. 3065-3084.

Coakley, Simon, Marian Gheorghe, Mike Holcombe, Shawn Chin, David Worth, and Chris Greenough (2012), "Exploitation of high performance computing in the FLAME agent-based simulation framework", in *2012 IEEE 14th International Conference on High Performance Computing and Communication and 2012 IEEE 9th International Conference on Embedded Software and Systems*, IEEE, pp. 538-545.

COMBINE (2018), *The COMBINE standards*, https:co.mbine.orgstandards.

Consortium, Gene Ontology (2018), *Gene ontology*, http://www.geneontology.org/.

Cooper RM Tsimring L, Hasty J (2017), "Inter-species population dynamics enhance microbial horizontal gene transfer and spread of antibiotic resistance", *eLife*, 6:e25950.

Cox, Jurgen and Matthias Mann (2011), "Quantitative, high-resolution proteomics for data-driven systems biology", *Annual review of biochemistry*, 80, pp. 273-299.

Cox, Robert Sidney, Curtis Madsen, James Alastair McLaughlin, Tramy Nguyen, Nicholas Roehner, Bryan Bartley, Jacob Beal, Michael Bissell, Kiri Choi, Kevin Clancy, et al. (2018), "Synthetic biology open language (SBOL) version 2.2. 0", *Journal of integrative bioinformatics*, 15, 1.

Crick, Francis (1970), "Central dogma of molecular biology", *Nature*, 227, 5258, p. 561.

Daghir-Wojtkowiak, Emilia, Pawel Wiczling, Malgorzata Waszczuk-Jankowska, Roman Kaliszan, and Michal Jan Markuszewski (2017), "Multilevel pharmacokinetics-driven modeling of metabolomics data", *Metabolomics*, 13, 3, p. 31, issn: 1573-3890, doi: 10.1007/s11306-017-1164-4, http://dx.doi.org/10.1007/s11306-017-1164-4.

Danos, Vincent and Sylvain Pradalier (2004), "Projective brane calculus", in *International Conference on Computational Methods in Systems Biology*, Springer, pp. 134-148.

Davidsen, Peter K., Nil Turan, Stuart Egginton, and Francesco Falciani (2016), "Multilevel functional genomics data integration as a tool for understanding physiology: a network biology perspective", *Journal of Applied Physiology*, 120, 3, pp. 297-309, issn: 8750-7587, doi: 10.1152/japplphysiol.01110.2014, eprint: http://jap.physiology.org/content/120/3/297.full.pdf.

Degano, Pierpaolo, Davide Prandi, Corrado Priami, and Paola Quaglia (2006), "Beta-binders for biological quantitative experiments", *Electronic Notes in Theoretical Computer Science*, 164, 3, pp. 101-117.

Delsanto, P.P., C.A. Condat, N. Pugno, A.S. Gliozzi, and M. Griffa (2008), "A multilevel approach to cancer growth modeling", *Journal of Theoretical Biology*, 250, 1, pp. 16-24, issn: 0022-5193, doi: http://dx.doi.org/10.1016/j.jtbi.2007.09.023.

Dematte, Lorenzo, Corrado Priami, and Alessandro Romanel (2008a), "Modelling and simulation of biological processes in BlenX", *ACM SIGMETRICS Performance Evaluation Review*, 35, 4, pp. 32-39.

— (2008b), "The BlenX language: a tutorial", in *International School on Formal Methods for the Design of Computer, Communication and Software Systems*, Springer, pp. 313-365.

Deutsch, Andreas, Sabine Dormann, et al. (2005), *Cellular automaton modeling of biological pattern formation*, Springer.

Di Carlo, Stefano, Gianfranco Politano, Alessandro Savino, and Alfredo Benso (2013), "A systematic analysis of a mi-RNA inter-pathway regulatory motif", *Journal of clinical bioinformatics*, 3, 1, p. 20.

Donatelli, Marcello and Andrea-Emilio Rizzoli (2008), “A design for framework-independent model components of biophysical systems”, in *International Congress on Environmental Modelling and Software.*

Donze, Alexandre (2010a), “Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems”, in *Computer Aided Verification: 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, ed. by Tayssir Touili, Byron Cook, and Paul Jackson, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 167-170, isbn: 978-3-642-14295-6, doi: 10.1007/978-3-642-14295-6_17, http://dx.doi.org/10.1007/978-3-642-14295-6%5C_17.

— (2010b), “Breach, a toolbox for verification and parameter synthesis of hybrid systems”, in *International Conference on Computer Aided Verification*, Springer, pp. 167-170.

Drasdo, D, A Buttenschon, and P Van Liedekerke (2018), “Agent-based lattice models of multicellular systems: numerical methods, implementation, and applications”, in *Numerical Methods and Advanced Simulation in Biomechanics and Biological Processes*, Elsevier, pp. 223-238.

Dubrova, Elena, Maxim Teslenko, and Andres Martinelli (2005), “Kauffman networks: Analysis and applications”, in *Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, IEEE Computer Society, pp. 479-484.

Duchene, Sebastian, David A. Duchene, Francesca Di Giallonardo, John-Sebastian Eden, Jemma L. Geoghegan, Kathryn E. Holt, Simon Y. W. Ho, and Edward C. Holmes (2016), “Cross-validation to select Bayesian hierarchical models in phylogenetics”, *BMC Evolutionary Biology*, 16, 1, p. 115, issn: 1471-2148, doi: 10.1186/s12862-016-0688-y, http://dx.doi.org/10.1186/s12862-016-0688-y.

Duncan, Craig, Kelvyn Jones, and Graham Moon (1998), “Context, composition and heterogeneity: using multilevel models in health research”, *Social science and medicine*, 46, 1, pp. 97-117.

Durand, Marc and Etienne Guesnet (2016), “An efficient Cellular Potts Model algorithm that forbids cell fragmentation”, *Computer Physics Communications*, 208, pp. 54-63.

Efroni, Sol, David Harel, and Irun R Cohen (2007), “Emergent dynamics of thymocyte development and lineage determination”, *PLoS computational biology*, 3, 1, e13.

Fisher, Jasmin and Thomas A Henzinger (2007), “Executable cell biology”, *Nature biotechnology*, 25, 11, p. 1239.

Fletcher, Alexander G, James M Osborne, Philip K Maini, and David J Gavaghan (2013), “Implementing vertex dynamics models of cell populations in biology within a consistent computational framework”, *Progress in biophysics and molecular biology*, 113, 2, pp. 299-326.

Gajski, Daniel D and Robert H Kuhn (1983), “New VLSI tools”, *Computer*, 12, pp. 11-14.

Galdzicki, Michal, Kevin P Clancy, Ernst Oberortner, Matthew Pocock, Jacqueline Y Quinn, Cesar A Rodriguez, Nicholas Roehner, Mandy L Wilson, Laura Adam, J Christopher Anderson, et al. (2014), “The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology”, *Nature biotechnology*, 32, 6, p. 545.

Gil-Quijano, Javier, Thomas Louail, and Guillaume Hutzler (2012), “From Biological to Urban Cells: Lessons from Three Multilevel Agent-Based Models”, in *Principles and Practice of Multi-Agent Systems: 13th International Conference, PRIMA 2010, Kolkata, India, November 12-15, 2010, Revised Selected Papers*, ed. by Nirmit Desai, Alan Liu, and Michael Winikoff, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 620-635, isbn: 978-3-642-25920-3, doi: 10.1007/978-3-642-25920-3_45.

Gilbert, W. (1991), “Towards a paradigm shift in biology”, *nat*, 349 (Jan. 1991), p. 99, doi: 10.1038/349099a0.

Gilks, Walter R, Sylvia Richardson, and David Spiegelhalter (1995), *Markov chain Monte Carlo in practice*, CRC press.

Gleeson, Padraig, Sharon Crook, Robert C Cannon, Michael L Hines, Guy O Billings, Matteo Farinella, Thomas M Morse, Andrew P Davison, Subhasis Ray, Upinder S Bhalla, et al. (2010), “NeuroML: a language for describing data driven models of neurons and networks with a high degree of biological detail”, *PLoS computational biology*, 6, 6, e1000815.

Godfrey, KR and JJ DiStefano III (1987), “Identifiability of model parameters”, *Identifiability of parametric models*, 1, pp. 1-20.

Goldberg, Robert N, Michael Cary, and Erek Demir (2010), “BioPAX A Community Standard for Pathway Data Sharing”, *Nature Biotechnology*, 28, Nature Biotechnology.

Green, Sara and Robert Batterman (2017), “Biology meets physics: Reductionism and multi-scale modeling of morphogenesis”, *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 61, pp. 20-34.

Gregor, Thomas (2017), “Beyond D’Arcy Thompson: Future challenges for quantitative biology”, *Mechanisms of development*, 145, pp. 10-12.

Guglielmi, G. and S. De Renzis (2017), “Optogenetic inhibition of apical constriction during Drosophila embryonic development”, in *Cell Polarity and Morphogenesis*, ed. by Thomas Lecuit, Methods in Cell Biology, Academic Press, vol. 139, pp. 167-186, doi: https://doi.org/10.1016/bs.mcb.2016.10.007, http://www.sciencedirect.com/science/article/pii/S0091679X16301509.

Gutenkunst, Ryan N, Joshua J Waterfall, Fergal P Casey, Kevin S Brown, Christopher R Myers, and James P Sethna (2007a), “Universally Sloppy Parameter Sensitivities in Systems Biology Models”, *PLOS Computational Biology*, 3, 10 (Oct. 2007), pp. 1-8, doi: 10.1371/journal.pcbi.0030189, http://dx.doi.org/10.1371/journal.pcbi.0030189.

— (2007b), “Universally sloppy parameter sensitivities in systems biology models”, *PLoS computational biology*, 3, 10, e189.

Hagberg, Aric, Pieter Swart, and Daniel S Chult (2008), *Exploring network structure, dynamics, and function using NetworkX*, tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Hallgrimsson, Benedikt and Brian K Hall (2011), *Epigenetics: linking genotype and phenotype in development and evolution*, Univ of California Press.

Harel, David (1987), “Statecharts: A visual formalism for complex systems”, *Science of computer programming*, 8, 3, pp. 231-274.

Harel, David and Amir Pnueli (1985), “On the development of reactive systems”, in *Logics and models of concurrent systems*, Springer, pp. 477-498.

Harel, David, Yaki Setty, Sol Efroni, Naamah Swerdlin, and Irun R Cohen (2008), “Concurrency in biological modeling: Behavior, execution and visualization”, *Electronic Notes in Theoretical Computer Science*, 194, 3, pp. 119-131.

Harris, Leonard A, Justin S Hogg, Jose-Juan Tapia, John AP Sekar, Sanjana Gupta, Ilya Korsunsky, Arshi Arora, Dipak Barua, Robert P Sheehan, and James R Faeder (2016), “BioNetGen 2.2: advances in rule-based modeling”, *Bioinformatics*, 32, 21, pp. 3366-3368.

Hoffmann, Hans-Peter (2012), “Deploying model-based systems engineering with IBM rational solutions for systems and software engineering”, in *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, IEEE, pp. 1-8.

Holzhutter, Hermann-Georg, Dirk Drasdo, Tobias Preusser, Jorg Lippert, and Adriano M. Henney (2012), “The virtual liver: a multidisciplinary, multilevel challenge for systems biology”, *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 4, 3, pp. 221-235, issn: 1939-005X, doi: 10.1002/wsbm.1158.

Houy, Constantin, Peter Fettke, and Peter Loos (2012), “Understanding Understandability of Conceptual Models–What Are We Actually Talking about?”, *Conceptual modeling*, pp. 64-77.

Huang, Sui (2012), “The molecular and mathematical basis of Waddington’s epigenetic landscape: A framework for post-Darwinian biology?”, *Bioessays*, 34, 2, pp. 149-157.

Hucka, Michael, Andrew Finney, Herbert M Sauro, Hamid Bolouri, John C Doyle, Hiroaki Kitano, Adam P Arkin, Benjamin J Bornstein, Dennis Bray, Athel Cornish-Bowden, et al. (2003), “The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models”, *Bioinformatics*, 19, 4, pp. 524-531.

Huddleston, Jennifer R (2014), “Horizontal gene transfer in the human gastrointestinal tract: potential spread of antibiotic resistance genes”, *Infection and drug resistance*, 7, p. 167.

Jensen, Kurt (1987), "Coloured Petri nets", in *Petri Nets: Central Models and Their Properties: Advances in Petri Nets 1986, Part I Proceedings of an Advanced Course*, ed. by W. Brauer, W. Reisig, and G. Rozenberg, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 248-299, isbn: 978-3-540-47919-2, doi: 10.1007/BFb0046842, https://doi.org/10.1007/BFb0046842.

Jernvall, Jukka, Stuart A Newman, et al. (2003), "Mechanisms of pattern formation in development and evolution", *Development*, 130, 10, pp. 2027-2037.

Joly-Guillou, M.-L. (2005), "Clinical impact and pathogenicity of Acinetobacter", *Clinical Microbiology and Infection*, 11, 11, pp. 868-873, issn: 1469-0691, doi: 10.1111/j.1469-0691.2005.01227.x, http://dx.doi.org/10.1111/j.1469-0691.2005.01227.x.

Kamada, Nobuhiko, Grace Y Chen, Naohiro Inohara, and Gabriel Núñez (2013), "Control of pathogens and pathobionts by the gut microbiota", *Nature immunology*, 14, 7, pp. 685-690.

Kimble, Judith (1981), "Alterations in cell lineage following laser ablation of cells in the somatic gonad of Caenorhabditis elegans", *Developmental biology*, 87, 2, pp. 286-300.

Kirk, Paul, Thomas Thorne, and Michael P H (2013), "Model selection in systems and synthetic biology." *Curr Opin Biotechnol*, 24, 4, pp. 767-74, doi: 10.1016/j.copbio.2013.03.012, http://dx.doi.org/10.1016/j.copbio.2013.03.012.

Koch, Ina (2015), "Petri nets in systems biology", *Software and Systems Modeling*, 14, 2, pp. 703-710.

Kohn, Dagmar and Nicolas Le Novere (2008), "SED-ML – An XML Format for the Implementation of the MIASE Guidelines", in *Computational Methods in Systems Biology*, ed. by Monika Heiner and Adelinde M. Uhrmacher, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 176-190, isbn: 978-3-540-88562-7.

Kong, Soonho, Sicun Gao, Wei Chen, and Edmund Clarke (2015), "dReach: delta-reachability analysis for hybrid systems", in *International Conference on TOOLS and Algorithms for the Construction and Analysis of Systems*, Springer, pp. 200-205.

Korem, Tal et al. (2015), "Growth dynamics of gut microbiota in health and disease inferred from single metagenomic samples", *Science*, 349, 6252, pp. 1101-1106.

Kummer, Olaf, Frank Wienberg, Michael Duvigneau, Jorn Schumacher, Michael Kohler, Daniel Moldt, Heiko Rolke, and Rudiger Valk (2004), "An extensible editor and simulation engine for Petri nets: Renew", in *International Conference on Application and Theory of Petri Nets*, Springer, pp. 484-493.

Kwiatkowska, Marta, Gethin Norman, and David Parker (2008), "Using probabilistic model checking in systems biology", *ACM SIGMETRICS Performance Evaluation Review*, 35, 4, pp. 14-21.

LACHOWICZ, MIROSLAW (2005), "Micro and meso scales of description corresponding to a model of tissue invasion by solid tumours", *Mathematical Models and Methods in Applied Sciences*, 15, 11, pp. 1667-1683.

Landauer, Rolf (1999), "Feynman and Computation", in ed. by Anthony J. G. Hey, Perseus Books, Cambridge, MA, USA, chap. Information is Inevitably Physical, pp. 77-92, isbn: 0-7382-0057-3, http://dl.acm.org/citation.cfm?id=304763.305684.

Lartillot, Nicolas, Thomas Lepage, and Samuel Blanquart (2009), "PhyloBayes 3: a Bayesian software package for phylogenetic reconstruction and molecular dating", *Bioinformatics*, 25, 17, pp. 2286-2288.

Le Novere, Nicolas, Michael Hucka, Huaiyu Mi, Stuart Moodie, Falk Schreiber, Anatoly Sorokin, Emek Demir, Katja Wegner, Mirit I Aladjem, Sarala M Wimalaratne, et al. (2009), "The systems biology graphical notation", *Nature biotechnology*, 27, 8, p. 735.

Lenoir, Timothy (1999), "Shaping biomedicine as an information science", in *Proceedings of the 1998 conference on the history and heritage of science information systems*, ASIS Monograph Series, Information Today, Inc., Medford, NJ, pp. 27-45.

Leonelli, S. (2012), "Introduction: Making sense of data-driven research in the biological and biomedical sciences", *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 43, 1, Data-Driven Research in the Biological and Biomedical

Sciences On Nature and Normativity: Normativity, Teleology, and Mechanism in Biological Explanation, pp. 1-3, ISSN: 1369-8486, DOI: http://dx.doi.org/10.1016/j.shpsc.2011.10.001, http://www.sciencedirect.com/science/article/pii/S136984861100077X.

LEY, RUTH E, DANIEL A PETERSON, and JEFFREY I GORDON (2006), "Ecological and evolutionary forces shaping microbial diversity in the human intestine", *Cell*, 124, 4 (Feb. 2006), pp. 837-48, DOI: 10.1016/j.cell.2006.02.017.

LI, MIN et al. (2008), "Symbiotic gut microbes modulate human metabolic phenotypes", *Proceedings of the National Academy of Sciences*, 105, 6, pp. 2117-2122.

LIEPE, JULIANE, PAUL KIRK, SARAH FILIPPI, TINA TONI, CHRIS P BARNES, and MICHAEL P H STUMPF (2014), "A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation", *Nat. Protocols*, 9, 2, pp. 439-456, DOI: http://dx.doi.org/10.1038/nprot.2014.025.

LIU, FEI and MONIKA HEINER (2010), "Colored Petri nets to model and simulate biological systems", *Recent advances in Petri Nets and concurrency*, 827, pp. 71-85.

LIU, FEI, MONIKA HEINER, and DAVID GILBERT (2017), "Coloured Petri nets for multilevel, multiscale and multidimensional modelling of biological systems", *Briefings in bioinformatics.*

LIU, FEI, MONIKA HEINER, and MING YANG (2016), "Fuzzy stochastic Petri nets for modeling biological systems with uncertain kinetic parameters", *PloS one*, 11, 2, e0149674.

LLOYD, CATHERINE M., MATT D.B. HALSTEAD, and POUL F. NIELSEN (2004), "CellML: its future, present and past", *Progress in Biophysics and Molecular Biology*, 85, 2, Modelling Cellular and Tissue Function, pp. 433-450, ISSN: 0079-6107, DOI: https://doi.org/10.1016/j.pbiomolbio.2004.01.004, http://www.sciencedirect.com/science/article/pii/S007961070400015X.

LOZUPONE, CATHERINE A, JESSE I STOMBAUGH, JEFFREY I GORDON, JANET K JANSSON, and ROB KNIGHT (2012), "Diversity, stability and resilience of the human gut microbiota", *Nature*, 489, 7415, pp. 220-230.

MACKLIN, DEREK N, NICHOLAS A RUGGERO, and MARKUS W COVERT (2014), "The future of whole-cell modeling", *Current Opinion in Biotechnology*, 28, pp. 111-115, ISSN: 0958-1669, DOI: http://dx.doi.org/10.1016/j.copbio.2014.01.012, http://dx.doi.org/10.1016/j.copbio.2014.01.012.

MARKOWETZ, FLORIAN (2017), "All biology is computational biology", *PLoS biology*, 15, 3, e2002050.

MARWAN, WOLFGANG, ANNEGRET WAGLER, and ROBERT WEISMANTEL (2011), "Petri nets as a framework for the reconstruction and analysis of signal transduction pathways and regulatory networks", *Natural Computing*, 10, 2, pp. 639-654.

MATTHEWS, LISA, GOPAL GOPINATH, MARC GILLESPIE, MICHAEL CAUDY, DAVID CROFT, BERNARD DE BONO, PHANI GARAPATI, JILL HEMISH, HENNING HERMJAKOB, BIJAY JASSAL, et al. (2008), "Reactome knowledgebase of human biological pathways and processes", *Nucleic acids research*, 37, suppl_1, pp. D619-D622.

MAUS, CARSTEN, STEFAN RYBACKI, and ADELINDE M UHRMACHER (2011), "Rule-based multi-level modeling of cell biological systems", *BMC Systems Biology*, 5, 1, p. 166.

MAYER, CHRISTIAN, CHRISTOPH HAFEMEISTER, RACHEL C BANDLER, ROBERT MACHOLD, RENATA BATISTA BRITO, XAVIER JAGLIN, KATHRYN ALLAWAY, ANDREW BUTLER, GORD FISHELL, and RAHUL SATIJA (2018), "Developmental diversification of cortical inhibitory interneurons", *Nature*, 555, 7697, p. 457.

MCKENZIE, KERRY (2017), "Ontic structural realism", *Philosophy Compass*, 12, 4, e12399.

MEHTA, PANKAJ, ALEX H LANG, and DAVID J SCHWAB (2016), "Landauer in the age of synthetic biology: energy consumption and information processing in biochemical networks", *Journal of Statistical Physics*, 162, 5, pp. 1153-1166.

MOLES, CARMEN G, PEDRO MENDES, and JULIO R BANGA (2003), "Parameter estimation in biochemical pathways: a comparison of global optimization methods", *Genome research*, 13, 11, pp. 2467-2474.

MOSCA, ETTORE, ROBERTA ALFIERI, IVAN MERELLI, FEDERICA VITI, ANDREA CALABRIA, and LUCIANO MILANESI (2010), "A multilevel data integration resource for breast cancer study", *BMC Systems Biology*, 4, 1, p. 76, ISSN: 1752-0509, DOI: 10.1186/1752-0509-4-76, http://dx.doi.org/10.1186/1752-0509-4-76.

Muganthan, Vinod A, Andrew Phillips, and Maria G Vigliotti (2008), "BAM: BioAmbient machine", in *2008 8th International Conference on Application of Concurrency to System Design*, IEEE, pp. 45-49.

Muggianu, F, A Benso, R Bardini, E Hu, G Politano, and S Di Carlo (2018), "Modeling biological complexity using Biology System Description Language (BiSDL)", in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, pp. 713-717.

Muggianu, Flavia (2018), *BiSDL - A new biological systems description language*, MA thesis, Politecnico di Torino.

Mussel, Christoph, Martin Hopfensitz, and Hans A Kestler (2010), "BoolNet—an R package for generation, reconstruction and analysis of Boolean networks", *Bioinformatics*, 26, 10, pp. 1378-1380.

Nakamura, Haruki, Gerard Kleywegt, Stephen K Burley, and John L Markley (2018), *Integrative Structural Biology with Hybrid Methods*, Springer.

Naldi, Aurelien, Celine Hernandez, Wassim Abou-Jaoude, Pedro T Monteiro, Claudine Chaouiya, and Denis Thieffry (2018), "Logical modeling and analysis of cellular regulatory networks with ginsim 3.0", *Frontiers in physiology*, 9.

Natale, Massimo, Alfredo Benso, Stefano Di Carlo, and Elisa Ficarra (2014), "FunMod: A Cytoscape Plugin for Identifying Functional Modules in Undirected Protein–Protein Networks", *Genomics, proteomics & bioinformatics*, 12, 4, pp. 178-186.

Nelson, Celeste M, Ronald P Jean, John L Tan, Wendy F Liu, Nathan J Sniadecki, Alexander A Spector, and Christopher S Chen (2005), "Emergent patterns of growth controlled by multicellular form and mechanics", *Proceedings of the National Academy of Sciences*, 102, 33, pp. 11594-11599.

Nepal, Chirag, Yavor Hadzhiev, Christopher Previti, Vanja Haberle, Nan Li, Hazuki Takahashi, Ana Maria M Suzuki, Ying Sheng, Rehab F Abdelhamid, Santosh Anand, et al. (2013), "Dynamic regulation of the transcription initiation landscape at single nucleotide resolution during vertebrate embryogenesis", *Genome research*, 23, 11, pp. 1938-1950.

North, Michael J, Nicholson T Collier, and Jerry R Vos (2006), "Experiences creating three implementations of the repast agent modeling toolkit", *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 16, 1, pp. 1-25.

Osborne, James M (2015), "Multiscale model of colorectal cancer using the cellular Potts framework", *Cancer informatics*, 14, CIN-S19332.

Oyama, Susan (2000), *The ontogeny of information: Developmental systems and evolution*, Duke university press.

Palm, Margriet M and Roeland MH Merks (2015), "Large-scale parameter studies of cell-based models of tissue morphogenesis using CompuCell3D or VirtualLeaf", in *Tissue morphogenesis*, Springer, pp. 301-322.

Palsson, Bernhard (2015), *Systems biology*, Cambridge university press.

Perry, Douglas L. (1993), *VHDL (2Nd Ed.)* McGraw-Hill, Inc., New York, NY, USA.

Plotkin, Gordon D (1981), "A structural approach to operational semantics".

Politano, Gianfranco, Alfredo Benso, Stefano Di Carlo, Francesca Orso, Alessandro Savino, and Daniela Taverna (2014), "A Computational Pipeline to Identify New Potential Regulatory Motifs in Melanoma Progression", in *International Joint Conference on Biomedical Engineering Systems and Technologies*, Springer, pp. 181-194.

Politano, Gianfranco, Alfredo Benso, Alessandro Savino, and Stefano Di Carlo (2014), "ReNE: a cytoscape plugin for regulatory network enhancement", *PloS one*, 9, 12, e115585.

Politano, Gianfranco, Federica Logrand, Mara Brancaccio, and Stefano Di Carlo (2017), "In-silico cardiac aging regulatory model including microRNA post-transcriptional regulation", *Methods*, 124, pp. 57-68.

Politano, Gianfranco, Francesca Orso, Monica Raimo, Alfredo Benso, Alessandro Savino, Daniela Taverna, and Stefano Di Carlo (2016), "CyTRANSFINDER: a Cytoscape 3.3 plugin for three-component (TF, gene, miRNA) signal transduction pathway construction", *BMC bioinformatics*, 17, 1, p. 157.

Politano, Gianfranco, Alessandro Savino, Alfredo Benso, Stefano Di Carlo, Hafeez Ur Rehman, and Alessandro Vasciaveo (2014), "Using Boolean networks to model post-transcriptional regulation in gene regulatory networks", *Journal of Computational Science*, 5, 3, pp. 332-344.

Pommereau, Franck (2015), "SNAKES: a flexible high-level petri nets library (tool paper)", in *International Conference on Applications and Theory of Petri Nets and Concurrency*, Springer, pp. 254-265.

Priami, Corrado and Paola Quaglia (2004), "Beta binders for biological interactions", in *International Conference on Computational Methods in Systems Biology*, Springer, pp. 20-33.

— (2005), "Operational patterns in Beta-binders", in *Transactions on Computational Systems Biology I*, Springer, pp. 50-65.

Purnick, Priscilla EM and Ron Weiss (2009), "The second wave of synthetic biology: from modules to systems", *Nature reviews Molecular cell biology*, 10, 6, p. 410.

Rajasundaram, Dhivyaa and Joachim Selbig (2016), "More effort — more results: recent advances in integrative 'omics' data analysis", *Current Opinion in Plant Biology*, 30, SI: 30: Genome studies and molecular genetics, pp. 57-61, issn: 1369-5266, doi: `http://dx.doi.org/10.1016/j.pbi.2015.12.010`.

Raman, Vasumathi, Alexandre Donze, Mehdi Maasoumy, Richard M Murray, Alberto Sangiovanni-Vincentelli, and Sanjit A Seshia (2014), "Model predictive control with signal temporal logic specifications", in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, IEEE, pp. 81-87.

Raykov, Tenko and George A Marcoulides (1999), "On desirability of parsimony in structural equation model selection", *Structural Equation Modeling: A Multidisciplinary Journal*, 6, 3, pp. 292-300.

Regev, Aviv, Ekaterina M Panina, William Silverman, Luca Cardelli, and Ehud Shapiro (2004), "BioAmbients: an abstraction for biological compartments", *Theoretical Computer Science*, 325, 1, pp. 141-167.

Richmond, Paul, Dawn Walker, Simon Coakley, and Daniela Romano (2010), "High performance cellular level agent-based simulation with FLAME for the GPU", *Briefings in bioinformatics*, 11, 3, pp. 334-347.

Riddle, Donald L, Thomas Blumenthal, Barbara J Meyer, and James R Priess (1997), *Developmental Genetics of the Germ Line–C. elegans II*, Cold Spring Harbor Laboratory Press.

Ritter, Frank E., Michael J. Schoelles, Karen S. Quigley, and Laura Cousino Klein (2011), "Determining the Number of Simulation Runs: Treating Simulations as Theories by Not Sampling Their Behavior", in *Human-in-the-Loop Simulations: Methods and Practice*, ed. by Ling Rothrock and S. Narayanan, Springer London, London, pp. 97-116, isbn: 978-0-85729-883-6, doi: `10.1007/978-0-85729-883-6\_5`, `https://doi.org/10.1007/978-0-85729-883-6%5C_5`.

Rodríguez-Rojas, Alexandro et al. (2013), "Antibiotics and antibiotic resistance: a bitter fight against evolution", *International Journal of Medical Microbiology*, 303, 6, pp. 293-297.

Said, H.M. (2018), *Physiology of the Gastrointestinal Tract*, Elsevier Science, isbn: 9780128124260, `https://books.google.it/books?id=2%5C_lQDwAAQBAJ`.

Sanfelice, Ricardo, David Copp, and Pablo Nanez (2013), "A toolbox for simulation of hybrid systems in Matlab/Simulink: Hybrid Equations (HyEQ) Toolbox", in *Proceedings of the 16th international conference on Hybrid systems: computation and control*, ACM, pp. 101-106.

Schaff, James C, Dan Vasilescu, Ion I Moraru, Leslie M Loew, and Michael L Blinov (2016), "Rule-based modeling with Virtual Cell", *Bioinformatics*, 32, 18, pp. 2880-2882.

Scheidel, Jennifer, Klaus Lindauer, Jorg Ackermann, and Ina Koch (2015), "Quasi-Steady-State Analysis based on Structural Modules and Timed Petri Net Predict System's Dynamics: The Life Cycle of the Insulin Receptor", *Metabolites*, 5, 4, pp. 766-793.

Schiff, Joel L (2011), *Cellular automata: a discrete view of the world*, John Wiley and Sons, vol. 45.

Schmid, Tobias and Alex Hajnal (2015), "Signal transduction during C. elegans vulval development: a NeverEnding story", *Current opinion in genetics and development*, 32, pp. 1-9.

Schreiber, Falk, Gary D Bader, Martin Golebiewski, Michael Hucka, Benjamin Kormeier, Nicolas Le Novere, Chris Myers, David Nickerson, Bjorn Sommer, Dagmar Waltemath, et al. (2015), "Specifications of standards in systems and synthetic biology", *Journal of integrative bioinformatics*, 12, 2, pp. 1-3.

Schulz-Trieglaff, Ole (2005), "Stochastic petri nets in systems biology", *BMC Bioinformatics*, 6, 3, P25.

Scott, Robert Jason, Hall Brian K., and Olson Wendy M. (2001), "Bridging the gap between developmental systems theory and evolutionary developmental biology†", *BioEssays*, 23, 10, pp. 954-962, doi: `10.1002/bies.1136`, eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/bies.1136`, `https://onlinelibrary.wiley.com/doi/abs/10.1002/bies.1136`.

Semenov, Sergey N, Albert SY Wong, R Martijn Van Der Made, Sjoerd GJ Postma, Joost Groen, Hendrik WH Van Roekel, Tom FA De Greef, and Wilhelm TS Huck (2015), "Rational design of functional and tunable oscillating enzymatic networks", *Nature chemistry*, 7, 2, p. 160.

Shahdad, Moe, Roger Lipsett, Erich Marschner, Kellye Sheehan, and Howard Cohen (1985), "VHSIC hardware description language", *Computer*, 18, 2, pp. 94-103.

Sharpe, James (2017), "Computer modeling in developmental biology: growing today, essential tomorrow", *Development*, 144, 23, pp. 4214-4225.

Shin, Hanna and David J Reiner (2018), "The Signaling Network Controlling C. elegans Vulval Cell Fate Patterning", *Journal of developmental biology*, 6, 4, p. 30.

Smith, Tony C and Eibe Frank (2016), "Introducing machine learning concepts with WEKA", *Statistical genomics: Methods and protocols*, pp. 353-378.

Sober, Elliott (1981), "The principle of parsimony", *The British Journal for the Philosophy of Science*, 32, 2, pp. 145-156.

Solovyev, Alexey, Maxim Mikheev, Leming Zhou, Joyeeta Dutta-Moscato, Cordelia Ziraldo, Gary An, Yoram Vodovotz, and Qi Mi (2010), "SPARK: a framework for multi-scale agent-based biomedical modeling", in *Proceedings of the 2010 Spring Simulation Multiconference*, Society for Computer Simulation International, p. 3.

Spiegelhalter, David J, Nicola G Best, Bradley P Carlin, and Angelika Van Der Linde (2002), "Bayesian measures of model complexity and fit", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 4, pp. 583-639.

Stephanou, Angelique and V. Volpert (2016), "Hybrid Modelling in Biology: A Classification Review", *Mathematical Modelling of Natural Phenomena*, 449, 1, pp. 37-48, doi: `10.1051/mmnp/201611103`.

Sternberg, P.W. (2005), *Vulval development (June, 25 2005), WormBook*, June 2005, `http:%5C/%5C/www.wormbook.org`.

Steyerberg, Ewout W, Andrew J Vickers, Nancy R Cook, Thomas Gerds, Mithat Gonen, Nancy Obuchowski, Michael J Pencina, and Michael W Kattan (2010), "Assessing the performance of prediction models: a framework for some traditional and novel measures", *Epidemiology (Cambridge, Mass.)* 21, 1, p. 128.

Sulston, John E and H Robert Horvitz (1977), "Post-embryonic cell lineages of the nematode, Caenorhabditis elegans", *Developmental biology*, 56, 1, pp. 110-156.

Sunnaaker, Mikael, Alberto Giovanni Busetto, Elina Numminen, Jukka Corander, Matthieu Foll, and Christophe Dessimoz (2013), "Approximate bayesian computation", *PLoS computational biology*, 9, 1, e1002803.

Svoboda, David, Tereza Necasova, Lenka Tesarova, and Pavel Simara (2018), "Tubular Network Formation Process Using 3D Cellular Potts Model", in *International Workshop on Simulation and Synthesis in Medical Imaging*, Springer, pp. 90-99.

Swat, Maciej H, Gilberto L Thomas, Julio M Belmonte, Abbas Shirinifard, Dimitrij Hmeljak, and James A Glazier (2012), "Multi-scale modeling of tissues using CompuCell3D", in *Methods in cell biology*, Elsevier, vol. 110, pp. 325-366.

Swerdlin, Naamah, Irun R Cohen, and David Harel (2008), “The lymph node B cell immune response: Dynamic analysis in-silico”, *Proceedings of the IEEE*, 96, 8, pp. 1421-1443.

Szklo, Moyses and Javier Nieto (2014), *Epidemiology*, Jones & Bartlett Publishers.

Tennent, Robert D. (1976), “The denotational semantics of programming languages”, *Communications of the ACM*, 19, 8, pp. 437-453.

Thomas, Christopher M and Kaare M Nielsen (2005), “Mechanisms of, and barriers to, horizontal gene transfer between bacteria”, *Nature reviews. Microbiology*, 3, 9, p. 711.

Tokishi, James and Yi-Chang Chiu (2013), “Evaluation and Improvement of Consistency of Hybrid and Multi-Resolution Traffic Simulation Models”, in *92nd Annual Meeting of the Transportation Research Board, Washington, DC.*

Trapnell, Cole, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn (2014), “The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells”, *Nature biotechnology*, 32, 4, p. 381.

Tronnolone, Hayden, Jennifer M Gardner, Joanna F Sundstrom, Vladimir Jiranek, Stephen G Oliver, and Benjamin J Binder (2017), “Quantifying the dominant growth mechanisms of dimorphic yeast using a lattice-based model”, *Journal of The Royal Society Interface*, 14, 134, p. 20170314.

Turnbaugh, Peter J et al. (2007), “The human microbiome project”, *Nature*, 449, 7164 (Oct. 2007), pp. 804-10, doi: `10.1038/nature06244`.

Uhrmacher, A. M., R. Ewald, M. John, C. Maus, M. Jeschke, and S. Biermann (2007), “Combining micro and macro-modeling in DEVS for computational biology”, in *2007 Winter Simulation Conference*, pp. 871-880, doi: `10.1109/WSC.2007.4419683`.

Uhrmacher, Adelinde M., Daniela Degenring, and Bernard Zeigler (2005), “Discrete Event Multilevel Models for Systems Biology”, in *Transactions on Computational Systems Biology I*, ed. by Corrado Priami, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 66-89, isbn: 978-3-540-32126-2, doi: `10.1007/978-3-540-32126-2\_6`, `http://dx.doi.org/10.1007/978-3-540-32126-2%5C_6`.

Valk, R (2003), “Object Petri nets: Using the nets-within-nets paradigm, Advanced Course on Petri Nets 2003 (J. Desel, W. Reisig, G. Rozenberg, Eds.), 3098”, *Appendix A: Proof of Theorem*, 3.

Vangheluwe, Hans LM (2000), “DEVS as a common denominator for multi-formalism hybrid systems modelling”, in *Computer-Aided Control System Design, 2000. CACSD 2000. IEEE International Symposium on*, IEEE, pp. 129-134.

Vasciaveo, Alessandro, Alfredo Benso, Stefano Di Carlo, Gianfranco Politano, Alessandro Savino, Fabrizio Bertone, Giuseppe Caragnano, and Olivier Terzo (2015), “A cloud-based approach for Gene Regulatory Networks dynamics simulations”, in *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, IEEE, pp. 72-76.

Von Stosch, Moritz, Joana Peres, Sebastiao Feyo de Azevedo, and Rui Oliveira (2010), “Modelling biochemical networks with intrinsic time delays: a hybrid semi-parametric approach”, *BMC systems biology*, 4, 1, p. 131.

Wade, Nicholas (2010), “Researchers say they created a ‘synthetic cell’”, *New York Times*, 20, A17.

Waltemath, Dagmar, Jonathan R Karr, Frank T Bergmann, Vijayalakshmi Chelliah, Michael Hucka, Marcus Krantz, Wolfram Liebermeister, Pedro Mendes, Chris J Myers, Pinar Pir, et al. (2016a), “Toward community standards and software for whole-cell modeling”, *IEEE Transactions on Biomedical Engineering*, 63, 10, pp. 2007-2014.

Waltemath, Dagmar, Jonathan R Karr, Frank T Bergmann, Vijayalakshmi Chelliah, Michael Hucka, Marcus Krantz, Wolfram Liebermeister, Pedro Mendes, Chris J Myers, Pinar Pir, Begum Alaybeyoglu, Naveen K Aranganathan, Kambiz Baghalian, Arne T Bittig, Paulo E Pinto Burke, Matteo Cantarelli, Yin Hoon Chew, Rafael S Costa, Joseph Cursons, Tobias Czauderna, Arthur P Goldberg, Harold F Gomez, Jens Hahn, Tuure Hameri, Daniel F Hernandez Gardiol, Denis Kazakiewicz, Ilya Kiselev, Vincent Knight-Schrijver, Christian Knupfer, Matthias Konig, Daewon Lee, Audald Lloret-Villas, Nikita Mandrik, J Kyle Medley, Bertrand Moreau, Hojjat Naderi-Meshkin, Sucheendra K Palaniappan, Daniel Priego-Espinosa, Martin Scharm, Mahesh Sharma, Kieran Smallbone, Natalie J Stanford, Je-Hoon Song, Tom Theile, Milenko Tokic, Namrata Tomar, Vasundra Toure, Jannis Uhlendorf, Thawfeek M Varusai, Leandro H Watanabe, Florian Wendland, Markus Wolfien, James T Yurkovich, Yan Zhu, Argyris Zardilis, Anna Zhukova, and Falk Schreiber (2016b), "Toward Community Standards and Software for Whole-Cell Modeling", *IEEE Trans Biomed Eng*, 63, 10 (Oct. 2016), pp. 2007-14, doi: 10.1109/TBME.2016.2560762, http://dx.doi.org/10.1109/TBME.2016.2560762.

Wang, Rui-Sheng, Assieh Saadatpour, and Reka Albert (2012), "Boolean modeling in systems biology: an overview of methodology and applications", *Physical biology*, 9, 5, p. 055001.

Wang, Wei-Dong, Zheng-Tang Chen, Bao-Guo Kang, and Rong Li (2008), "Construction of an artificial intercellular communication network using the nitric oxide signaling elements in mammalian cells", *Experimental cell research*, 314, 4, pp. 699-706.

Wilson-Kanamori, John, Vincent Danos, Ty Thomson, and Ricardo Honorato-Zimmer (2015), "Kappa rule-based modeling in synthetic biology", in *Computational Methods in Synthetic Biology*, Springer, pp. 105-135.

Witten, M, BL Bodnar, and AC Liu (1987), "Simulation and modeling of a single bus tightly coupled multiprocessor system", *Mathematics and computers in simulation*, 29, 1, pp. 19-31.

Wolf, Yuri I, Mikhail I Katsnelson, and Eugene V Koonin (2018), "Physical foundations of biological complexity", *Proceedings of the National Academy of Sciences*, 115, 37, E8678-E8687.

Wright, Thomas and Ian Stark (2018), "The Bond-Calculus: A Process Algebra for Complex Biological Interaction Dynamics", *arXiv preprint arXiv:1804.07603*.

Xie, Wangang, Paul O Lewis, Yu Fan, Lynn Kuo, and Ming-Hui Chen (2010), "Improving marginal likelihood estimation for Bayesian phylogenetic model selection", *Systematic biology*, 60, 2, pp. 150-160.

Zhao, P, L Zhang, JA Grillo, Q Liu, JM Bullock, YJ Moon, P Song, SS Brar, R Madabushi, TC Wu, et al. (2011), "Applications of physiologically based pharmacokinetic (PBPK) modeling and simulation during regulatory review", *Clinical Pharmacology and Therapeutics*, 89, 2, pp. 259-267.