

Software implementations allowing new approaches
toward data analysis, modeling and curation of
biological knowledge for Systems Medicine

Doctoral thesis

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Biology

John Zobolas

Trondheim, May 2021

Contents

Acknowledgements	v
Abstract	vii
Paper list	xi
Primary	xi
Additional	xii
Abbreviations	xiii
Summary	1
One for all	1
Dots and lines	3
Knowledge from a stack of papers	5
Biological Dictionaries aid in the curation of complex knowledge	8
Sharing causal interactions with PSICQUIC	11
Biological modeling: a Prelude	15
Clean Code	18
Get the right rules	24
Links to software, documentation and data analyses	29
GitHub organizations	29
Documentation	29
DrugLogics software modules	29
R community packages	30
Miscellaneous data analyses and repositories	30
References	31
Papers	43

Acknowledgements

I first acknowledge funding from the **ERACoSysMed project COLOSYS**. Now to people, the most important part: **I would like to thank everyone** that interacted with me during the years I conducted my PhD research (2017-2021), even for a tiny bit. A person is only a small node in a complex network of interactions, which only when considered together, make up the person. In other words, you are not just you! So, you were **all important** for me (and for others I am sure)! Importance though is measured in varying degrees. Therefore now, I will proceed to give some personal credits (the part that most of you came here to read :)

A big **THANK YOU** to my supervisor, Prof. Martin Kuiper. He gave me the opportunity to come to Norway, which opened a world of possibilities. The dream supervisor everyone should have and a caring human being above all. A true gentleman and an unsurpassed cook as well!

To Dr. Åsmund Flobak for excellent co-supervision, the nice scientific discussions we had and for introducing me into R, which largely influenced the way I do science. Spending time with his lively family was always a nice change of pace. That day we were all together at Amsterdam's zoo and I saw my first penguins, will remain unforgettable!

To Dr. Steven Vercruyse for our scientific discussions and for teaching me and ins and outs of web software development. I will never forget the ELIXIR Biohackathon 2020, great laughs and great work, all in one. And of course, I will never forget the “tour” . Thanks Steven!

To Prof. Astrid Laegreid, for suggesting to me to participate in the Responsible Research and Innovation (RRI) course, which inspired me to think more broadly about my research and the world we live in, and of course meet several wonderful people! That's also how I became acquainted with Digital Life Norway (DLN). Liv Eggset Falkenberg did an excellent job at coordinating the DLN Research School and she was co-organizer of the Walkshop in Jotunheimen (September 2019), which was a truly wonderful experience. With DLN, I had the benefit of participating in various conferences across Norway and the opportunity to do an industry internship in Sweden during the cold winter of 2021, so thanks DLN and Liv!

To Noemi Del Toro Ayllon, for introducing me to the professional world of software development and project management with Java. Visiting the IntAct team at EBI during the summer of 2018 was a memorable experience and when she came to Trondheim later in 2019, we had such a great time, so thank you Noe!

To Henning Hermjacob, for not just being the scientific host for my visits to EBI in England,

but also for hosting me in his lovely Airbnb house each time! Spending a few months in Cambridge during my PhD was a truly marvelous experience, so thanks Henning!

To Professors Denis Thieffry, Pedro T. Monteiro and Jin-Dong Kim for the nice discussions we had, resulting in excellent collaborations.

And of course, **to my colleagues from the DrugLogics group**, for the good times we spent inside and outside of work! I am especially grateful to Barbara Niederdorfer and Evelina Folkesson for our music sessions. Eirini Tsirvouli has been a very positive, dynamic presence. Rafel Riudavets Puig has been a really close friend - I hope that in the future we get to continue our random walks that somehow always end up in McDonalds! Marcio Luis Acencio has been a good friend as well, with a wonderful family that gained two new members I got to meet before he left our group! Vasundra Touré has been a wonderful colleague, a true source of light for the time we spend at our office in Glosshaugen (and it was really dark in general). Wine, cheese, standards and good memories! Also, favorite cafe buddies with Anamika Chatterjee - we certainly made Espresso House richer (though I later switched to Dromedar, say sorry)!

Last but not least, a big **THANKS** to the beautiful city of Trondheim! I've had some really inspirational walks in these historic roads. And to its nice coffee shops I've been working throughout my PhD! Diverse working environment is extremely important and as it was perfectly stated:



John Zobolas, May 2021

Abstract

Cancer is one the most prevalent human diseases. The scientific community has devoted considerable efforts to understand the mechanisms behind this disease and search for treatments that promise a better quality of life for patients. To accomplish this goal, Biology and Medicine have joined forces with Computer Sciences, using the power of Computational modeling, Mathematics, Machine Learning and Statistics. This interdisciplinary effort to address the cancer problem, constitutes the basis upon which this thesis was formed. We present several contributions to this effort, consisting of software, data analyses and mathematical investigations, which have enabled the more efficient curation of biological knowledge, the use of computer models to prioritize drug treatments for cancer and the derivation of molecular mechanistic insights from the simulation results.

In order to build a computational model of a biological system such as a cancer cell, we first need a way to describe the structure of such a system. A common network-based approach provides an elegant representation of such a structure, where molecular entities such as proteins and genes are connected to each other via causal interactions, which in turn determine cellular behavior and the functional properties of the cell as an integrated system of individual components. These interactions form the Prior Knowledge Network (PKN), which serves as the basic building block for most computational biological models. Nonetheless, several challenges exist, even at this early stage of the modeling process.

The first problem is that biological information by its very nature is largely complex, and therefore its formalization to a structured, computable form for use in modeling applications, demands extra attention. The translation of scientific knowledge from publications into such a computable form is achieved with the use of specialized software tools and is the main responsibility of biocurators. In order to help biocurators be more efficient in their annotation tasks, we proposed the Visual Syntax Method (VSM) as an alternative approach for general-purpose knowledge formalization. In particular, we implemented a user interface component (VSM-box) that enables curators to annotate any type of information, no matter its complexity, and translate it into an intuitive, flexible sentence-like format. This software was used to build a prototype curation interface (CausalBuilder) for the annotation of molecular causal interactions, which constitute the cornerstone of a model's PKN.

The second problem concerns the availability and ease of access of causal molecular interaction data for modeling or other scientific endeavors. A standard format for the representation of such signaling information was developed (CausalTAB), and we supported the export of the causality statements from CausalBuilder’s interface to this format. But there exist several other molecular interaction databases that could update their data to fit the new CausalTAB standard. PSICQUIC is a web-service platform that was initially built so that users can conveniently fetch in a standard way molecular interaction data from different sources. We extended PSICQUIC to incorporate the new CausalTAB format, so that causality-enriched information generated by our curation prototype tool or from other data providers could be shared through a common channel.

A third major problem arose during the design process of the VSM-box and its application, CausalBuilder. Behind the scenes, the curator interface has to communicate with a large number of diverse biological data resources, each with its own online API service that provides access to the respective data. In order to present to the user the available terms that pertain to a specific annotation of interest, a uniform way to query all these resources was needed. This prompted us to build the Unified Biological Dictionaries (UBDs), a software suite that provides a unified gateway for life science data, helping users retrieve the right query terms. In addition, curators sometimes come across new knowledge that is not yet available through the standard authoritative resources. To address this related problem, we connected UBDs with PubDictionaries, an online resource of simple dictionaries, allowing curators to publicly create and share ad-hoc terms, and further use them as annotations in VSM-based applications.

After the signaling information has been curated and the causal interactions assembled to form the PKN, we then need to specify the mathematical equations of the cancer cell model. This allows us to describe and analyze its dynamical behavior subject to external stimuli, such as drug perturbations. The modeling approaches can in general range from qualitative to quantitative and in this work we focused on Boolean modeling, where signaling components are assigned either an active or inactive state. An automated computational pipeline was developed to produce an ensemble of Boolean models from a PKN, calibrated to a specific cancer cell signaling phenotype. These models are then analyzed to suggest possible synergistic drug combinations and the results are compared with experimental findings, where all possible combinations are tested in a high-throughput screen setup. We demonstrated that our pipeline could prioritize specific drug combinations, reducing the number of drugs that need to be tested in experiments, before a viable treatment is found for a patient. Moreover, several analyses indicated that our models can be used to derive mechanistic insights about the diseased model and generate novel biological hypotheses. Lastly, we showed the significance of the PKN quality, where even small modifications to the cancer signaling network could severely affect our pipeline’s drug prediction performance.

To exploit the range of parameterizations present in the Boolean models produced by our pipeline, we devised several strategies to split and compare the different models in a dedicated R package (*emba*). This supplementary effort allowed us to find potential biomarkers, which are nodes whose state is decisive for the global behavior of the models and can indicate parts of the PKN that are responsible for a drug combination to be synergistic. Additionally, we noticed particular patterns in the way specific equations always correspond to specific signaling states in our models, so we more deeply investigated the influence of the choice of parameterization on the output behavior of these nodes. This led us to propose a list of Boolean function metrics that can assist modelers in choosing more appropriate equations, meaning those that are consistent with the regulatory information present in the PKN and whose expected output better matches experimental observations. Finally, results from a study of diverse Boolean functions indicated that these also exhibit diverse output behaviors, with some being highly biased towards specific Boolean outcomes while others depending more on the ratio between positive and negative regulators, as these are derived from the two distinct types of causal interactions present in the model's PKN.

Paper list

Primary

Papers that I am first author:

- **Paper 1:** *UniBioDicts: Unified access to Biological Dictionaries*

Zobolas, J., Touré, V., Kuiper, M., & Vercruyse, S. [1]

SV and VT identified the need for software to connect with the resources listed in MI2CAST used in the CausalBuilder tool. JZ implemented the software and wrote the manuscript. All co-authors revised and provided inputs to the manuscript.

- **Paper 2:** *Linking PubDictionaries with UniBioDicts to support Community Curation*

Zobolas, J., Kim, J.-D., Kuiper, M., & Vercruyse, S. [2]

MK and SV developed the original idea for this project. JZ wrote the application for a BioHackathon project and invited JDK to collaborate. JZ implemented the client side, JDK the server side. JZ wrote the manuscript. All co-authors revised and provided inputs to the manuscript.

- **Paper 3:** *Fine tuning a logical model of cancer cells to predict drug synergies: combining manual curation and automated parameterization*¹

Flobak, Å., **Zobolas J.**, Vazquez M., Steigedal T., Thommesen L., Grislingås A., Niederdorfer B., Folkesson E., Kuiper M.

AF designed the project, developed initial software and executed experiments. JZ extended the software, ran all simulations, produced and analyzed results. AF and MK added biological interpretation and wrote the manuscript. JZ provided various inputs and text to the manuscript. TS, LT and AG helped AF with experiments. MV developed prototype software. BN and EF did curation work and performed experiments.

¹(Manuscript) Shared first co-authorship, to be submitted to the Molecular Systems Biology Journal

- **Paper 4:** *emba: R package for analysis and visualization of biomarkers in Boolean model ensembles*

Zobolas, J., Kuiper, M., & Flobak, Å. [3]

AF developed the idea of this project. JZ wrote the software and the manuscript. All co-authors revised and provided inputs to the manuscript.

- **Paper 5:** *Boolean function metrics can assist modelers to check and choose logical rules*²

Zobolas, J., Monteiro, P. T., Kuiper, M., & Flobak, Å. [4]

JZ designed this project and wrote the manuscript. PTM and AF provided feedback and ideas to better shape the content of the manuscript. All co-authors revised and provided inputs to the manuscript.

Additional

In the following papers I have contributed to the underlying software and manuscript text:

1. *VSM-box: General-purpose Interface for Biocuration and Knowledge Representation*
Vercruyse, S., **Zobolas, J.**, Touré, V., Andersen, M. K., & Kuiper, M. [5]
2. *CausalBuilder: Bringing the MI2CAST Causal Interaction Annotation Standard to the Curator*
Touré, V., **Zobolas, J.**, Kuiper, M., & Vercruyse, S. [6]
3. *CausalTAB: the PSI-MITAB 2.8 updated format for signalling data representation and dissemination*
Perfetto, L., Acencio, M. L., Bradley, G., Cesareni, G., Del Toro, N., Fazekas, D., Hermjakob, H., Korcsmaros, T., Kuiper, M., Lægreid, A., Lo Surdo, P., Lovering, R. C., Orchard, S., Porras, P., Thomas, P. D., Touré, V., **Zobolas, J.**, & Licata, L. [7]

²(Preprint) To be submitted to the Journal of Theoretical Biology

Abbreviations

abmlog	All possible Boolean Models Link Operator Generator
AGS	Gastric Adenocarcinoma (cell line)
API	Application Programming Interface
AUC	Area Under the Curve
BioPax	Biological Pathway Exchange
CASCADE	CAncer Signaling CAusality DatabasE
CNA	Copy Number Alterations
COVID-19	COrona VIrus Disease 2019
drabme	Drug Response Analysis to Boolean Model Ensembles
emba	Ensemble (Boolean) Model Biomarker Analysis
gitsbe	Generic Interactions To Specific Boolean Equations
GO	Gene Ontology
GREEKC	Gene Regulation Ensemble Effort for the Knowledge Commons
HUPO-PSI	HUMAN Proteome Organization - Proteomics Standards Initiative
IMEx	The International Molecular Exchange Consortium
MI2CAST	Minimum Information about a Molecular Interaction CAusal STatement
MIQL	Molecular Interactions Query Language
MITAB	Molecular Interaction TABular format
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PKN	Prior Knowledge Network
PPI	Protein-Protein Interaction
PRC	Precision Recall Curve
PSICQUIC	Proteomics Standard Initiative Common QUery InterfaCe
ROC	Receiver Operating Characteristic
SBGN	Systems Biology Graphical Notation
TF-TG	Transcription Factor - Target Gene

UBDs (UniBioDicts)	Unified Biological Dictionaries
UMAP	Uniform Manifold Approximation and Projection (for dimension reduction)
VSM	Visual Syntax Method
XML	Extensible Markup Language

Summary

One for all

Scientific and technological progress has been the foundation for some of the most astounding achievements of humankind. In the last century in particular, discoveries were made that contributed to the sustainable development of the economy and society, affecting our lives in an unprecedented manner and making possible what was considered impossible. The invention of the digital computer and the Internet for example, revolutionized the access, dissemination and analysis of information [8,9]. We have been to the Moon, a breakthrough that has opened up the possibilities of space exploration and interstellar travel. The industrial revolution of the latest century has enabled us to design machines for every conceivable need. Human well-being has become significantly better: compare a middle class household and the appliances within, with one from 60 years ago. With a higher standard of living and the ongoing efforts to alleviate hunger, poverty and inequality on a global scale, people have started caring more about the planet, paving the way for sustainable economic and environmental growth [10]. Due to advancements in Biology and Medicine, the application of public health interventions such as vaccinations and hygiene measures has become common practice, causing a rapid increase in the global life expectancy during the last century [11]. The genome-editing technology CRISPR [12] has enabled the discovery of new therapeutic solutions for a variety of genetic diseases and has been beneficially used in several agriculture and plant biotechnology applications [13]. The list of achievements is truly endless and all the data points to the fact that the world is getting better [14].

There are three factors that have made technological progress possible. Firstly, every human innovation is based on basic scientific research, without which the development of new technologies would have been impossible. Secondly, society is developing new contracts with science [15], where researchers can only be trusted to continue their work (and get funding for it), if they tackle real-world problems and produce knowledge characterized by a fully transparent and participative spirit. Practically, this means that better communication skills are a necessity for today's scientists and that their research should have translational potential to deliver on society's expectations. But solving these real-world problems is incredibly hard, and so, they cannot be addressed by applying

knowledge from specific fields only, e.g. either from the Computer or the Biological Sciences alone. This brings us to the third factor: in order for science to deliver on its promises to society, collaboration across fields of science is the only way forward.

Medicine, from research to develop new therapies up to delivering the actual product or services to the patients, constitutes the perfect example that encompasses all three reasons that have enabled progress to transpire in its domain. It first starts with a real-life problem: people get sick. The existence of diseases is a societal problem and a hard one at that, since people usually lack the necessary knowledge or the means to deal with it on their own. They have in fact exchanged some of their freedom to have a place in society, and ensure that they receive proper treatment when needed (along with other forms of security, access to free education, etc.). To manage such a complex problem, society provides healthcare services, which have significantly increased across the world in recent years [16]. For most people, the single most applied healthcare interaction is the use of drugs, prescribed by medical doctors. Drugs are the translational product of the pharmaceutical industry, which is the result of basic interdisciplinary research. Medical doctors alone wouldn't be able to find the cause and understand the mechanisms behind many of the diseases that exist today. This knowledge has been the culmination of years of scientific knowledge, built atop collaboration across fields, from Medicine and Biology to Computer Science and Engineering.

So, only by using every possible method and knowledge at our disposal and by working together, we can achieve the solution to complex problems such as human diseases. When these conditions are met, societal challenges can be addressed and science stands as one unified body for the good and progress of all mankind. The field of Systems Medicine has been the direct embodiment of this notion, promising improved prevention, prognosis, diagnosis and treatment of patients via an integrative, interdisciplinary approach [17].

Dots and lines

One of the simplest ways to conceptualize complex systems, either man-made or existing freely in nature, is using the notion of a *network* or *graph*. The idea is that any system is composed of individual entities or components of interest (nodes) and these components interact with each other in various, usually non-obvious ways (links). These two properties, namely having some objects to study, and relationships between these objects, form the basis for the conceptualization of a network (Figure 1). From a cognitive point of view, the conceptualization of a network manifests as a visual representation in our brain, consisting of a bunch of dots (nodes) connected with numerous lines (links) [18]. Such a projection is usually close to what people instinctively draw on paper when they attempt to describe their knowledge about a system and its inner workings (thereby “connecting the dots”). Simple schematics that are abstractly similar to dots and lines, along with further contextual information (e.g. node labels, coloring, directed links, etc.), seem to be able to capture and render information derived from our thought processes, in a unique and comprehensible way.

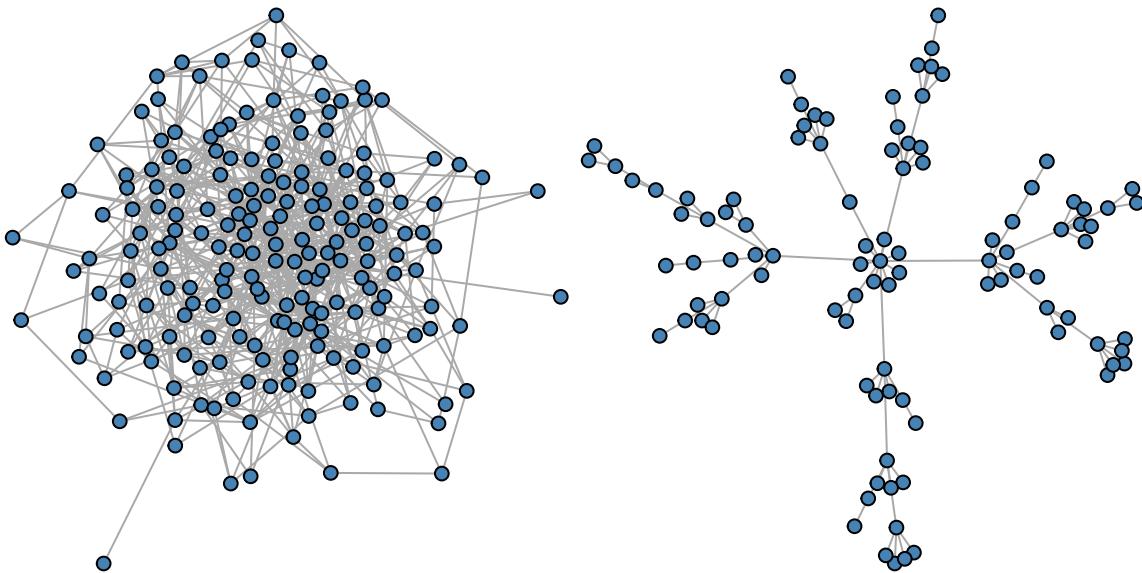


Figure 1: Two examples of networks, composed of dots and lines. The left network is a random graph based on the Erdős-Rényi model [19] and the one on the right is created using the preferential attachment principle that characterizes scale-free networks with hub nodes, such as the World Wide Web [20].

Since studying complex systems falls into the domain of science’s responsibilities, and graphs seem to be an intuitive way of representing such systems, the emergence of a new field called network science was inevitable [21]. Its purpose is to establish a unified set of tools and methods to study the properties of any type of network that emerges across disparate fields. A variety of software tools for network visualization and analysis have been released throughout the years, ranging from generic-purpose [22–25], to tools more suitable for studying biological [26–29] or social networks [30,31]. The use of such tools enables the discovery of fundamental

laws that characterize the function of systems represented by networks. In addition, it allows us to study in detail the networks' systemic structure and derive key principles that drive their evolution and emergent behavior. Anthropological research for example uses network theory to study people and their relationships, and explain emergent complex phenomena such as human behavior. Neuroscience uses network analysis methods to detect anomalies in diseased human brains [32]. The impact of online social networks is studied to understand and predict future personal and profit-oriented communication (online marketing) [33]. Epidemiologists use graph-based methods to model the spread of diseases like COVID-19, predict the future course of outbreaks and evaluate strategies to control epidemics [34]. Molecular biologists study intra- and intercellular signaling networks to understand the mechanisms behind biological processes and investigate the causes of network dysregulation, often leading to the emergence of particular disease phenotypes. Such network-based approaches have significant clinical applications since they have the potential to assist in the discovery of new disease genes and modules, and the identification of drug targets and biomarkers for complex diseases [35].

The work presented in this thesis is heavily based on this network medicine paradigm, with causal molecular interaction networks as the main object of study. Our primary focus is on protein-protein interaction (PPI) networks, with proteins as nodes and their physical contacts and interactions as links, and gene regulatory networks, represented for example by directed regulatory relationships between transcription factors and genes (TF-TG networks). These types of networks demonstrate a system of signal transduction pathways connected by crosstalk and embedded in feedback loops, forming what is known as the *Prior Knowledge Network* (PKN). The causality property of the PKN stems from the fact that the network links are directed (i.e. protein X affects protein Y) and signed (Y is inhibited or activated as a result). It's exactly this causality information that allows the investigation of behaviors from a systems perspective. Such networks form the basis for the study and computational modeling of cancer, which is another subject of investigation in this thesis. In the subsequent chapters, we will discuss how we addressed problems related to the formalization, access and public sharing of the knowledge encoded in the PKN.

Knowledge from a stack of papers

Where does the information that is used to build knowledge networks originate from? One of the most widely adopted ways to record and share knowledge, has been the publication of scientific findings in specialized journals. This has resulted in a major challenge that researchers in the life sciences face, which is to stay updated with the huge amount of information that is published on a daily basis (Figure 2). It becomes impossible for the average scientist to find, read, extract and use that information in an efficient manner without the use of databases. Even when using databases, one is often confronted with both chronically incomplete knowledge, and also a lack of sufficient contextual information to assess when exactly the knowledge is valid.

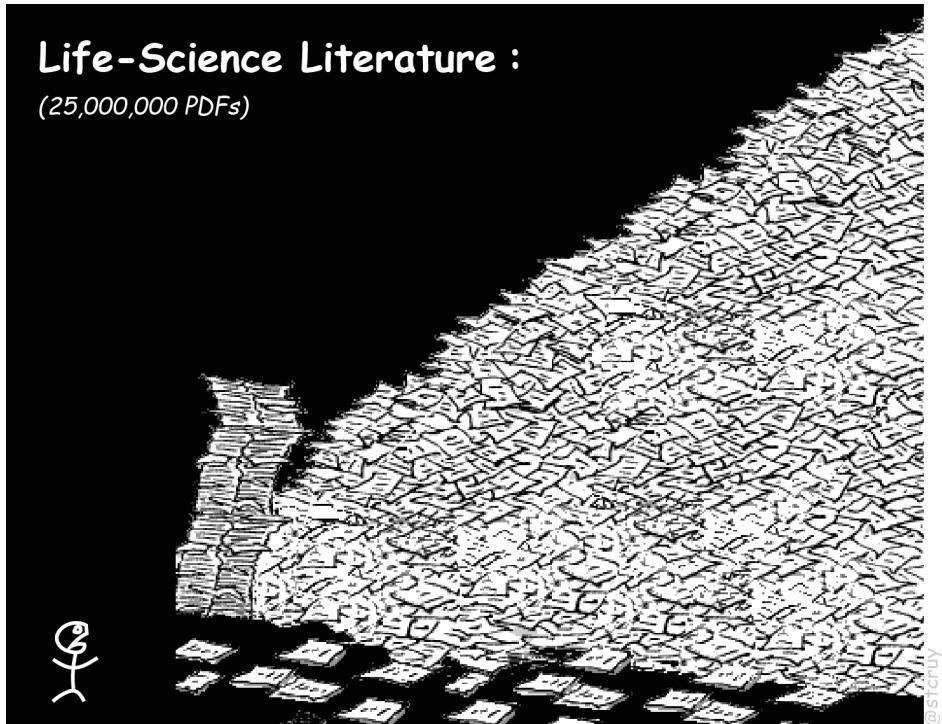


Figure 2: Human vs Life-Science Literature. How can humans stay up-to-date with increasing knowledge stored in PDF files? [36]

A severe problem lies already at the data entry stage. Biocurators are people whose main task is to read the scientific literature and translate knowledge into a precise, computable form, ready to be inserted into databases [37,38]. The huge body of literature existing today is full of inconsistencies and inaccuracies, so expert interpretation and annotation are essential. But current databases are limited in what they can contain, because there exists no easy way to properly transfer all kinds of complex knowledge or ideas into them, in the first place. Moreover, the annotation tools that biocurators use are not intuitive nor flexible enough to be used by large crowds of people, to convert vast amounts of relevant knowledge from the scientific literature into the respective databases. The insufficient funding to curate scientific results into databases, and the cost of creating

a new knowledge base for every new project, are some extra confounding factors. Because of this, researchers all over the world have to spend considerable time performing ad-hoc manual curation of publications that are relevant for their project, often with improvised approaches (Word, Excel). At best they also spend time developing a specialized curation platform or computational methods to extract knowledge, which can only capture a fraction of the “actual reported truth” [39]. Nonetheless, all these efforts form a significant part of the scientific enterprise, assisting in the creation of digital knowledge repositories, which are subsequently used to build PKNs for the computational modeling of biological processes.

A list of tools have been created to assist biocurators in their annotation tasks. Notably, the IntAct editor is an open-source desktop application software that enables IntAct curators and members of the IMEx consortium to annotate molecular interactions [40]. Because of the lack of installation instructions and documentation, coupled with a complex interface, specialized training from senior IntAct curators is required to learn how to use this software. Nonetheless, it is one of the most used and effective tools for the job, since it has been around for a lot of years and during that time, there has always been a spirit of close collaboration between developers and curators to implement features, solve bugs and in general improve the annotation capabilities of the software. Canto is another tool that was built to support community curation in the PomBase fission yeast database [41]. It has now expanded its original purpose to support curation of other model organism databases and different molecular data types (e.g. annotation of a larger set of GO terms). Canto’s respective website provides extensive documentation and step-by-step user guidance throughout the annotation procedure [42]. A user management mechanism is incorporated in the software so as to allow proper monitoring of curation tasks and efficient communication between curators for work prioritization. In addition, two relatively new tools have been developed for the curation and visualization of molecular interaction maps: NaviCell [43] and MINERVA [44]. These tools facilitate knowledge exploration in addition to knowledge annotation, allowing for an interactive user experience (e.g. feedback via comments), enabling content sharing, supporting well known data standards (e.g. SBGN [45]) and thus allowing for data interoperability and re-use. All the aforementioned annotation tools are limited by the fact that they aren’t generic enough to curate any type of information, with most of them representing specialized solutions pertaining to specific annotation purposes. Most tools require extra technical configurations and software to include additional levels of contextualized details required for current and future curation efforts.

To obtain support from computational pipelines that will help us process the vast amounts of knowledge and advance our understanding of processes in nature, we must be able to efficiently annotate and store information that is highly detailed and contextualized. Hereby, the knowledge's inherent complexity should be kept manageable and understandable by humans and machines alike. In order to accommodate for a much more powerful, flexible, and reusable annotation process, an intuitive curation and knowledge formalization method was developed, called VSM (Visual Syntax Method) [46]. VSM enables scientists to capture any type of knowledge with any type of contextual information, in a way that is understandable by both humans and computers.

Part of the work in this thesis has been to assist in the implementation of a software module that implements VSM as a general-purpose, web-based user interface, named VSM-box [5]. This software component was used to build CausalBuilder, a prototype curation interface for the annotation of causal molecular interactions [6]. CausalBuilder uses VSM to generate concrete, customizable templates that represent causal statements. It supports the export of the annotated statements in standard signaling formats, such as CausalTAB [7], which can be stored in relevant databases or used to build computational models of biological processes. To support the large variety of contextual information related to causal molecular interactions between biological entities, allowing for a finer disambiguation between seemingly similar or conflicting causality statements (e.g. a transcription factor simultaneously up and down regulating a target gene in different cellular contexts), CausalBuilder was designed to comply with a list of guidelines (MI2CAST) that were developed exactly for this purpose [47]. All in all, CausalBuilder provides biologists and curators with a simple user interface for the annotation of causal regulatory knowledge, translating highly contextual information about molecular interactions from scientific publications to a computable form.

Biological Dictionaries aid in the curation of complex knowledge

During the design process of the VSM-box tool and its application, CausalBuilder, we came across a critical technical issue that needed to be addressed, and whose resolution had ramifications outside of the intended scope of our work. Due to the high degree of complexity within the domain of biology, biocurators need to annotate diverse information, taken from a plethora of biological resources and vocabularies. To enable a wider expressiveness in the annotation of causal statements, the recommended list of ontologies and vocabularies of the MI2CAST standard had to be rather extensive [48]. Since CausalBuilder conforms to the MI2CAST standard, a unified way to retrieve, format and display vocabulary terms from different databases was needed. We illustrate this with an example: in Figure 3, a simple VSM-template that a curator can use to annotate a causal statement with CausalBuilder is shown. Following the MI2CAST guidelines, the source entity of the causal statement (first box in Figure 3) must always be specified and a list of recommended resources where the annotation could potentially originate from is provided [48]. We limit the number of these resources to three in this example, making it so that the source biological entity can be annotated as a protein (from UniProt [49]), a complex (from Complex Portal [50]), or an RNA transcript (from RNACentral [51]). The intended use case is that the curator will type in a string (e.g. “tp53”) and a list of terms and descriptive metadata from the three respective standard databases will be returned. This information can be displayed by VSM-box in an autocomplete drop-down menu to ease the selection of the appropriate term by the curator.

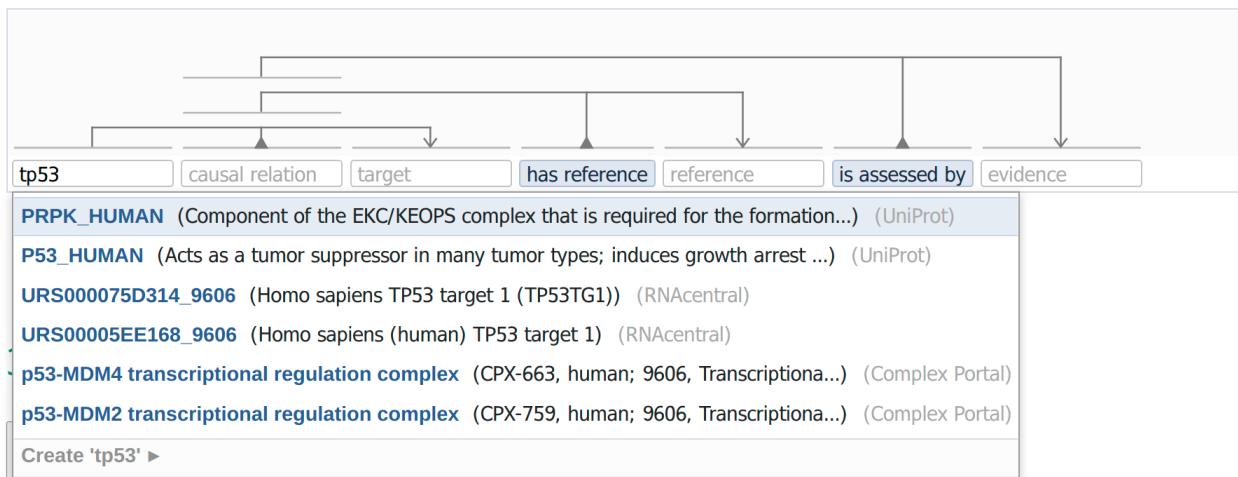


Figure 3: Querying multiple data resources using the VSM-box technology in CausalBuilder. The user enters a string of interest and selects a list of resource types (not shown here) for the source entity, following the MI2CAST curation guidelines. The UBDs stand as a hidden translator between the query launched from the curator interface and the respective database data, returning a list of uniformly-structured matches, shown as a drop-down list to the user. The matches consist of a curator-friendly main term (shown in blue) and metadata like identifier, name of species, textual description, resource name etc., that a user can use to disambiguate between the different concepts.

We can now clearly state the heart of the problem: the resources that offer protein, RNA and multiprotein complex data, have different online APIs to serve their information, and it is usually structured in diverse formats. Therefore, it was necessary to design a generic solution that would translate all the necessary information from the recommended resources of the MI2CAST standard into a unified representation schema. Then, we could implement modules that “talk” to the databases and translate the provided information into this uniform data format. As a result of having a standardized way to represent data from various disparate resources, VSM-box and other curation tools could easily process the returned data load and create drop-down menus to help users in their annotation tasks (as shown in Figure 3). The outcome of all this effort was the implementation of UBDs (Unified Biological Dictionaries, see [Paper 1](#)). The reason for the name *dictionaries* originates from the abstract data type called *associative array* (also known as *map* or *dictionary*), which is a collection of (*key*, *value*) pairs, and is an integrated feature of many programming languages. For our application, we reasoned that the minimum information that is needed for the unique identification of concepts for curation tasks is a computer-friendly ID and a human-friendly term, precisely matching the *key* and *value* of the associative array’s data structure.

An unforeseen consequence of the UBDs implementation was that by covering most of the vocabularies and ontologies recommended by the MI2CAST standard, we ended up mapping into a unified format a large amount of diverse terminologies across life sciences. This happened because our solution encapsulated and extended other similar efforts, such as the BioPortal [52] and EBI Search [53] web services. We therefore managed to bring even more biomedical ontologies and biological data resources under one umbrella, and subsequently increase the accessibility, interoperability and reusability of the provided data [54]. So, even though UBDs main user is the software engineer building curation tools (as we were at the beginning of this effort with CausalBuilder), several computational researchers can benefit from our implementation, if they need to query disparate biological resources for lightweight information (i.e. terms, identifiers and some metadata) using a single programmatic interface. In the end, the feedback we got from biocurators who tested the CausalBuilder tool was very encouraging, pointing out that we had proceeded in the right direction with our efforts to build UBDs, the hidden machinery enabling all the autocomplete “magic” to happen in VSM-box’s user interface.

The implementation of UBDs put us in a position to confront problems that biocurators face during their annotation tasks, which haven’t yet been properly addressed by any existing technology. One of these challenges is that biocurators often need to annotate terms in a specific domain or novel field, for which there is still no authoritative database or ontology nor a community consensus about the respective terminology [55]. A similar challenge manifests when new knowledge is discovered or similarly, further contextual information related to existing knowledge comes into light, as a result of scientists’ constant drive for progress. This eventually leads to the constant refactoring of ontologies and identifiers, subsequently making biocurators life even more difficult.

To respond to these challenges, biocurators create project-specific, *ad-hoc* vocabularies that aren't openly accessible and usually become obsolete after some time passes. We reasoned that with the UBDs infrastructure in place, we could do better.

In summary, the core of the problem is two-fold: first, biocurators need a simple way to annotate new information that does not yet exist in any resource and second, this information needs to be shared publicly for further review and management by expert communities. To tackle this problem, we collaborated with experts from PubDictionaries, an online repository of publicly accessible and editable dictionaries [56]. Using the online interface of PubDictionaries, curators can create simple dictionaries, consisting of terms and identifiers of their own choice, solving the second part of the problem. Additionally, by updating the PubDictionaries API and connecting all existing and future public dictionaries with UBDs and their underlying unified format, we streamlined their use in annotation tools and solved the first part of the problem. The technical work was carried out during an intense hacking week at the ELIXIR Biohackathon 2021 event and the implementation details are described in [Paper 2](#) of this thesis. As a final result, we showcased a demo in which curators could use their public, ad-hoc terminologies from PubDictionaries, to annotate a simple sentence using the VSM-box interface.

Sharing causal interactions with PSICQUIC

Defining standards is a very important initiative across scientific disciplines, since it facilitates the accessibility and sharing of information amongst data users as well as the interoperability of software tools that produce or process the respective data, thereby increasing the quality of research findings [54]. Following this logic, after the curation of causal molecular interactions from scientific literature has been achieved, the next step is to store this information to a standard data format. One of the most detailed, community-standard formats for representing molecular interaction data is the PSI molecular interaction (MI) XML format, released by the Human Proteome Organization Proteomics Standards Initiative (HUPO-PSI) [57]. The newest version of this standard is the PSI-MI XML 3.0 [58]. A simplified format for interaction data was provided by the same organization, called the Molecular Interaction Tabular format (PSI-MITAB). PSI-MITAB has become popular amongst the scientific community since it is more user-friendly and Microsoft Excel-compatible, compared to the respective XML-based format [59]. PSI-MITAB version 2.7 in particular, encapsulated many details of interest regarding a molecular interaction in a total of 42 columns, but it did not include information about its causality. This resulted in an effort to standardize the signaling information and subsequent vocabulary terminology for causal molecular interactions, originally led by SIGNOR's database curators [60]. The new PSI-MITAB version 2.8 (also called **CausalTAB**), included four new columns to incorporate additional details related to a molecular interaction's directionality (defining the biological roles of the regulator and target, one column each), regulatory mechanism (e.g. indirect causal regulation or post-translational modification) and resulting effect (up or down regulation of the target) [7].

After having a signaling data format for molecular interactions in place, the next challenge was to find a way to share such information efficiently. The heart of the problem was the same even without the addition of causality information: a large number of molecular interaction databases exist, each one with different APIs to access the respective data. Since no single database can incorporate the totality of molecular interactions pertaining to a specific biological system of interest, users have to collect data from diverse databases by launching queries in different websites or by directly downloading the respective files, which might not always be offered in standardized formats. To ease the computational access and retrieval of molecular interaction data from various resources, a web service with a common query interface (PSICQUIC) and language (MIQL) was developed [61]. Using the PSICQUIC web service, users can now download all relevant data files from their databases of interest in different PSI-MI compliant formats, suitable for further analysis (Figure 4).

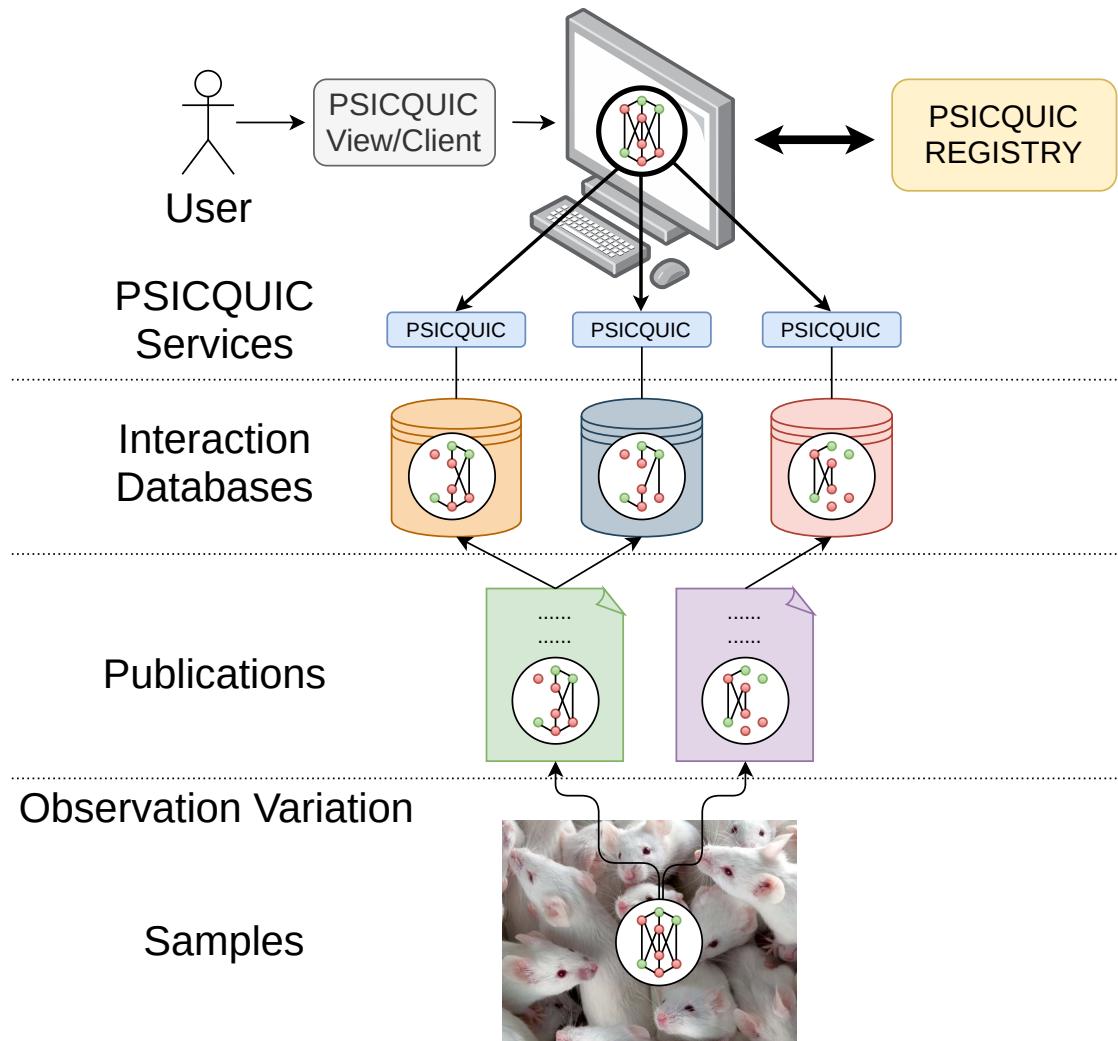


Figure 4: PSICQUIC architecture. Molecular interaction knowledge about a biological system, supported by different experimental methods, is being reported in publications. Each of these publications reports part of the actual truth about the studied system. This knowledge is curated from the respective publications and inserted to diverse molecular interaction databases. The databases share their data in standard formats (e.g. PSI-MITAB) via the PSICQUIC web service and are part of a registry list. Users launch queries via a PSICQUIC web client to retrieve the distributed molecular interaction data and synthesize the complete observed knowledge of the studied system, suitable for further analysis and visualization.

With a new signaling data format established by the relevant scientific community and the PSICQUIC web service contributing to the accessibility of molecular interaction data, the next step was to support the PSI-MITAB 2.8 format in PSICQUIC. A software project to extend the PSICQUIC platform and include causality information of molecular interactions was thus formed.³ Our contribution to this effort was the development of the underlying PSICQUIC software (version 1.4) that indexes CausalTAB files provided by the respective data providers, enabling the query

³This project was funded by the GREEKC (Gene Regulation Ensemble Effort for the Knowledge Commons) COST action (<https://www.greekc.org/>) and was realized as a Short Term Scientific Mission (STSM) in cooperation with engineers from the IntAct team at the European Bioinformatics Institute (EBI).

and subsequent download of causality-enriched interactions. The PSICQUIC View website source code was updated to show the four new columns in the HTML table results. Additionally, several clients that enable programmatic access to the distributed signaling data (written in Java, Python and Perl programming languages), were refactored to comply with data fitting the new standard format. Lastly, the relevant PSICQUIC documentation was improved and reformatted to enhance user readability [62].

The aforementioned development effort spurred a series of actions that led to several improvements in the PSICQUIC platform. The Molecular Interactions Community, is an open source community providing tools, standard formats, ontologies and modules for manipulating molecular interaction data [63]. For example, some of these modules are used to read and write PSI-MITAB files across different versions. Since these modules hadn't been updated for years (showing signs of software rot [64]), we had to refactor the codebase and add tests to ensure it's future reliability and quality. In the end, even though we managed to complete the task and support CausalTAB in PSICQUIC via updating these modules, the need to replace them with a newer library was imminent. JAMI is such a library, integrating all standard molecular interaction data formats such as the PSI-MI XML and PSI-MITAB, into a unified implementation, hiding the complex details of each format from the developers and thereby making their work easier [65]. The implementation work was initiated in a GREEKC workshop [66] and continued during the first ELIXIR BioHackathon in Paris [67]. Upon finishing the support for CausalTAB in JAMI,⁴ we provided the first PSICQUIC service indexing SIGNOR's CausalTAB data at the time, made available through a development server [68].

During the ELIXIR BioHackathon, the architecture details of a new cloud-based, distributed PSICQUIC service were discussed and documented for future development efforts. The goal is to enable the data providers to upload their molecular interaction data in a fully automated process and add support for data validation. This service will minimize the long-term commitment and maintenance from the data providers, which they cannot always afford (e.g. deployment of the server hosting PSICQUIC). Another outcome of the BioHackathon was the draft implementation of a new PSICQUIC View interface, aiming to modernize and update the current web application used to access PSICQUIC [69]. Further work needs to be done to import and use newer technologies in the interface, which will result in better filtering and sorting of the HTML table results and more interactive, graph-based visualizations of the PSICQUIC data. To broadly facilitate the sharing of causal interaction data, additional development efforts are needed, in particular towards improving existing PSICQUIC clients. One such example is the PSICQUIC Universal Client, a Cytoscape app for querying multiple PSICQUIC-compliant interaction data services from a simple user interface [70]. This client has been used in tutorials to guide novice users in the visualization and analysis of molecular interaction networks [71]. Lastly, two more clients that need to be updated to support

⁴During the GREEKC Marseille Hackathon 2019 event, see more info on the project here: https://github.com/GREEKC/hackathon-marseille/tree/master/project_descriptions/causal_psicquic

the latest CausalTAB signaling format are the PSICQUIC [72] and PItools R packages [73]. These packages enable the translation of molecular interaction data directly into formats suitable for computational analysis with R, and therefore are crucial for relevant computational tasks.

Biological modeling: a Prelude

In previous chapters we summarized our efforts related to the curation, access and sharing of causal molecular interactions, the building block of PKNs. This is only the first step in the process of creating computational models of biological systems [74]. To translate signaling networks to a computational form for simulation purposes, the next step is to define the regulatory *rules (parameterization)* of the underlying system: how do the different network entities influence each other across time and potentially space, and how does the systemic behavior change when specific rules or parameters of the model are modified? [75] The formulation of the rules in mathematical terms (i.e. they are in most cases expressed as equations), defines the mechanistic details of the studied models and thus enables the study of their dynamics. Practically, this means that models can be used for various forms of analyses and simulations, and their outputs further investigated. For example, models can be validated and tested for agreement with experimental observations, can make predictions and generate new hypotheses leading to the design of new experiments, and they can be subjected to various perturbations, with their subsequent effects on the studied systems quantified and thoroughly analyzed. In addition to the biological analysis, one of the more important aspects of mathematical modeling is that it enables the investigation of the underlying mechanisms that result in the manifestation of the described system's behavior. In other words, computational models are explainable and interpretable, enabling us to answer why things happen the way they do, which is one of the driving forces behind science itself.

Several mathematical methodologies have been used to study biological systems, such as stochastic modeling, ordinary and partial differential equations (ODE and PDE modeling), Petri nets, logical modeling, Bayesian networks, cellular automata and agent-based modeling [76]. Each modeling paradigm encapsulates a different way of formalizing the underlying rules and makes different assumptions about the studied system. Such assumptions can for example include the temporal and spatial properties of the modeled system (space and time can each be treated in a continuous manner or with varying degrees of discretization), the molecular scale (focus on modeling individual molecules or discrete amounts of each molecule or just considering their molar concentrations) and the nature of interactions between the molecules (reaction processes can be described as happening in a stochastic or deterministic manner). In general, there is a trade-off between the complexity of the system that a model is constructed to simulate and the mechanistic detail incorporated in the model itself [75]. Ideally we would like to have models which can simulate highly complex systems in as much detail as possible. This poses a significant challenge, since a more detailed representation of a biological system requires a higher level of granularity inherent in a model's formalization. This means that more parameters are required to specify and calibrate the model for the simulation and accurate representation of reality, and larger amounts of experimental data and computational resources are also necessary. On the other hand, by sacrificing the complexity

of the studied system and making simpler models, we face overwhelming uncertainties that need to be properly quantified and integrated in any interpretation of results from such a model [77]. In the end, the modeling scope is a crucial factor for the choice of the appropriate methodology, and thus sufficient knowledge of the advantages and disadvantages of each formalism can be beneficial towards selecting the approach deemed most suitable for the realization of the modeling objectives.

In this thesis we focus on Boolean modeling, one of the simplest formalizations for the modeling of complex biological systems [78]. In this type of qualitative approach, every individual entity has a binary state denoting activity (1) or inactivity (0) and time is discretized [79]. Every interaction that affects a target entity is assembled into a Boolean equation which defines that particular target's output activity in the next time step. To formulate such a Boolean equation, only knowledge of the regulatory network topology is needed, along with the use of logical operators that describe how the combined activity of the regulators affects the target. This inherent simplicity in defining the rules is what makes the Boolean formalism attractive to modelers. Moreover, since the PKN is one of the core elements that characterize the Boolean rules, this explains why we spent a large amount of our efforts in this thesis to make sure modelers get the proper contextual prior knowledge. Lastly, another advantage of Boolean modeling is that it does not require the specification of parameters such as kinetic rate constants and initial concentrations that are a strong prerequisite in other modeling formalisms (e.g. in ODE modeling), where there is always a need for large and expensive amounts of data that might be either lacking or not enough to adequately characterize the rules.

Continuing with the explanation of the modeling formalism, a logical model is a list of mathematical equations expressed in Boolean algebra. The state of such a system is represented by a series of 0's and 1's, each corresponding to the activity state of a signaling entity. Using the Boolean rules, we can update the system's state by deciding on the order that each of its equations are applied, to derive the next entity states. Therefore, a synchronous update scheme can be defined as calculating the output of all Boolean equations of the model at the same time. In contrast, randomly specifying one or more equations to update can result in various forms of asynchronous dynamics, which enable the inclusion of processes with different time scales in a logical model. By repeatedly applying the Boolean rules, systemic states that either don't change (*fixpoints*) or ones that demonstrate cycling patterns, can be reached. These are the *attractors*, which represent solutions to the system of equations that constitute a Boolean model and their identification is synonymous to the study of the long-term dynamical behavior of the modeled system. Attractors have been shown to be biologically meaningful, either by representing specific phenotypic outputs [80] or transitions between system states like in the cell cycle [81].

Several computational tools have been developed to aid in the dynamical analysis of Boolean models [82]. These tools enable users to easily create and edit logical models, identify different types of attractors and their reachability properties, analyze model state evolution over time,

investigate phenotypic outputs subject to various types of perturbations, explore different model parameterizations and calibrate models to fit experimental data, among others [83]. The existence of such a plethora of tools has enabled the modeling of complex diseases, the discovery of potential therapeutic solutions and the investigation of biomarkers that correlate with patients' response to specific pharmaceutical drugs. In particular, the derivation of mechanistic insights related to the manifestation of diseases, is one of the main challenges that computational modeling efforts strive to address towards achieving the goal of personalized medicine [84]. In light of this, several logical modeling approaches have been used to stratify patients based on the integration of multi-omics data [85], build patient-specific models that aid in the understanding of drug sensitivity and cancer resistance mechanisms [86–88] and help identify novel therapeutic targets [89–91]. Part of the work in this thesis has been to complement the aforementioned approaches by developing a software pipeline that uses causal prior knowledge and tailors Boolean models to cell-specific cancer signaling activities (**Paper 3**). These models can subsequently be used to predict combinatorial treatments that aid in the prioritization of drugs in high-throughput screening technologies and will eventually provide better clinical decision support for cancer patients, helping us find optimal drug-patient matches.

Clean Code

Our main goal is to use causal molecular knowledge and signaling activity data to build and parameterize Boolean models to represent specific cancer cell systems, and study their behavior in the presence of in-silico drug perturbations. As such, the importance of the underlying scientific software that enables these computational tasks, is unquestionable. Such software needs to satisfy a list of requirements pertaining to its suitability for practical use [92]. Such practices ensure that the software does what it is intended to do and its produced scientific simulations can be used to inform decision-making for clinical applications. In other words, for the scientific results of the simulations to be *actionable* and *trustworthy* [93], the following software requirements are not just optional, but rather a necessity. At first, challenges related to automation, efficiency and optimization in terms of the computational resources needed to perform the simulations and analyses, need to be properly addressed. To promote collaboration and allow others to study, use and further develop the software, the respective codebase needs to be open sourced [94,95]. Standard formats for input and output should be supported, as well standard libraries for common programming tasks, enabling the effortless integration with related software. Also, sufficient documentation needs to be provided, containing installation guidelines and explanations for the various configurations used in the simulations [96]. Simple examples of use and related tutorials should be part of such documentation as well [97]. The usability of the software can also be increased by applying better programming architecture and design principles (e.g. writing modular code), which also makes the software easier to test, extend and verify. Making the results of the simulations verifiable (i.e. by ensuring that the algorithmic procedures and the model equations are *right*), will increase their reproducibility, further supporting the aforementioned goals [98]. All in all, there can only be gains if the code is *clean* and properly taken care of [99].

In Figure 5 we present an overview diagram of the DrugLogics software pipeline. This pipeline represents a computational software system aiming to assist in the identification of synergistic drug combinations. The pipeline's two main modules, Gitsbe and Drabme, are both written in Java. Gitsbe, using as input a PKN and a signaling activity profile for some of the key entities in the input network, creates Boolean models based on a genetic algorithm approach and calibrates them to best fit the training signaling activity data. Boolean models with higher fitness have fixpoint attractors, whose node states best match the binary signaling activities. Calibration refers to changes in the Boolean equations to produce such high fitness models. These changes can simply be variations in the parameterization such as mutations in the Boolean rules (e.g. a logical OR becomes a logical AND and vice-versa) or topological changes, such as the addition or removal of a regulator's effect on a particular target. Thus the Boolean models in Gitsbe's produced ensemble are parameterized differently, but are all essentially approximate representations of the same biological system. These variant Boolean models are then used as input to Drabme, to test the effect of several perturbations

from a given drug panel and quantify their combinatorial interaction patterns using well-known synergy frameworks (namely HSA [100] and Bliss [101]). Leveraging the wisdom of the crowds [102] using Gitsbe’s models, Drabme’s output predictions contribute in reducing the exponentially large drug space that is associated with combinatorial treatments for the targeted therapy of cancer, providing a list of candidate combinations to test in the lab, before a viable solution is found for a patient.

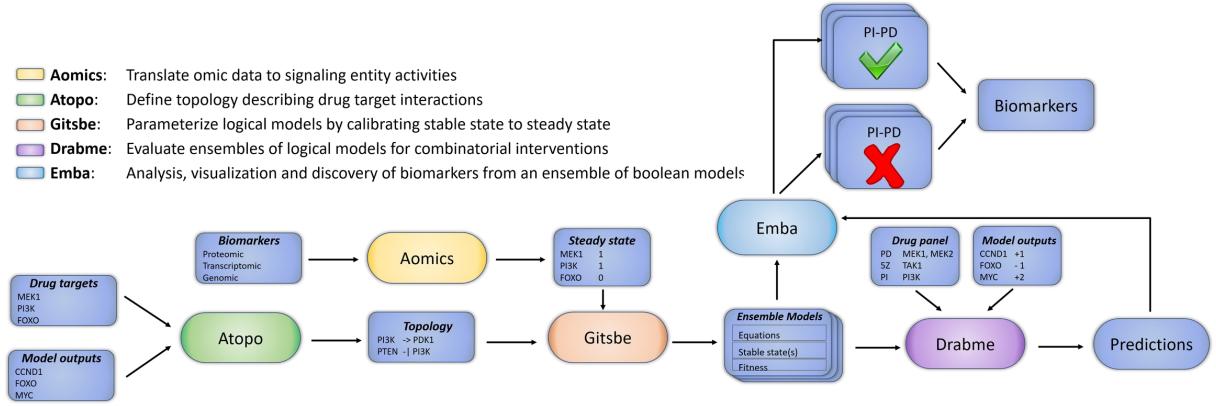


Figure 5: The DrugLogics software pipeline. A series of connected modules that build a regulatory topology incorporating specific drug targets, parameterize Boolean models to a specific cancer signaling profile assembled from various omics data and simulate drug combination perturbations. The output models and their predictions can be further analyzed to explain the difference between phenotypes and thus identify biomarkers that make a particular drug combination synergistic.

The development of Drabme and Gitsbe was part of the latest stages of a previous research thesis and it was mostly exploratory work [103]. To lay the groundwork for the analyses and investigations of the papers included in this thesis, additional software development needed to be done. We started by refactoring the source code to increase its readability, maintainability and extensibility. Java classes were restructured, variables and functions were renamed to reflect current best programming practices, code written from scratch for common tasks such as string manipulation was replaced with reusable Java components (e.g. using libraries from Apache Commons [104]) and the Maven project management tool [105] was used to enable easier source compilation, installation and packaging (all the Java code was bundled in a single compressed file for ease of use). To increase interoperability of our software, we used the Java library BioLQM [106] to enable the export of the produced Gitsbe models to standardized formats in the logical modeling community, such as GINML (used in the GINsim software [107]), SBML-qual [108] and BoolNet [109]. For the calculation of fixpoint attractors, an external scripting-based tool was used by default [110]. We added a built-in, integrated Java solution using BioLQM, which apart from fixpoints can also identify minimal trapspaces, a generic type of attractor that allows for a deeper exploration of dynamics.

A practical outcome of these efforts was that the code became more modular, enabling the addition of software tests using the JUnit5 framework [111] and specialized libraries [112,113]. With more tests, hidden or otherwise impossible to pinpoint bugs, were identified and fixed. Some of these were critical for the validity of the output findings, since they related to how changes in the parameterization or topology were encoded in the software equivalent of a Boolean model’s equations. Moreover, it became much easier to add new features to the software, e.g. we supported the execution of parallel simulations in Gitsbe (a significant performance optimization) and incorporated a new synergy framework for the identification of synergistic drug combinations in Drabme (Bliss). Gitsbe’s simulation is the core computational process that produces the “best-fit” models resulting from the evolutionary approach of the genetic algorithms. Selecting a few of those best-fit models at the end of each simulation and executing multiple simulations in parallel, each one associated with a different random seed number, is what generates a reproducible list of Boolean models for use in Drabme (or other software if models are exported to standard formats). Lastly, we shared publicly the developed modules in GitHub⁵ and built an extensive online documentation using the R package bookdown [114] to gather all related information in one place with regard to the software’s configuration parameters, the mathematical calculations used, installation instructions and examples of use.⁶ This online documentation became a central virtual hub, providing information on all the software modules in the pipeline. One of these modules was `druglogics-synergy`, a convenient Java package used to serially execute Gitsbe and Drabme in one go, and which was employed for the simulations of [Paper 3](#).

With all the main code in proper order, we could start investigating the outputs of our software and assess the performance of the produced drug synergy predictions. The results of these efforts, along with biologically-relevant mechanistic insights derived from further analyzing the simulation data and performing various investigations, are analytically presented in [Paper 3](#), and in even more detail at the `ags-paper` [repository](#), which also includes reproducibility guidelines. In the following paragraphs we are going to briefly explain the input data and software that either needed to be in place before we started the experimentation with the in-silico simulations or was built to help further analyze the output Boolean models from Gitsbe and Drabme’s synergy predictions.

To begin with, we had to choose a reference drug combination dataset to compare our predictions against. The argument here is to use a dataset that you know very well so that you can first experiment with the software and its configurations, and only later test your predictions on other published (and potentially larger) datasets. Therefore, we chose the Flobak et al. (2019) dataset, with a total of 153 combinations of 18 targeted drugs, involving measurements across 8 cancer cell lines [115]. Next, we needed to assess in a computational manner which of the drug combinations in the reference dataset (per cell line) are synergistic and which are not. Our

⁵See respective repositories [here](#)

⁶See documentation repositories [here](#)

first attempt was to manually check the output growth curves and derive a majority-assessed *gold standard* (so it was more of a curative group effort). We continued by performing a thorough analysis using the CImbinator tool [116] and established a methodology which computed the synergy classification of the reference dataset that best matched our internal curation efforts to call synergy (see [sintef-obs-synergies repository](#)). All in all, we had a reference dataset and a list of drug combinations designated as synergistic from it, ready to be used to evaluate Drabme’s predictions on the same dataset.

On another front, we also needed a PKN suitable for our analysis. In particular, the targets of the drugs used in the reference dataset should be entities in the network, so as to enable the simulation of drug perturbations in Gitsbe’s derived Boolean models. Moreover, several of the most important pathways in cancer cell biology (e.g. PI3K, ERK and TGFB signaling [117]) would have to be included as well. A PKN that fits all the above characteristics was curated within our research group and refined throughout many years, resulting in the topology that was used for the simulations of [Paper 3](#) (named [CASCADE](#)). In addition, proper signaling data was needed to train the Gitsbe models to a cancer proliferating phenotype. For that purpose, we used the literature curated activity profile for a set of nodes in [CASCADE](#), which was the result of a previous research work [90]. This activity profile concerns only one of the cell lines in the reference dataset, namely the gastric adenocarcinoma cell line (AGS). In summation, by employing a curated topology and training data from only one cell line, we could focus more on the model parameterization aspects of our software, the performance assessment of the synergy predictions and the rest of the investigations of [Paper 3](#).

While exploring various configuration options for the Gitsbe and Drabme simulations,⁷ we needed a tool to quickly assess their effect on the pipeline’s performance and see which were the most important to tune. The pipeline’s performance here refers to Drabme’s output synergy predictions (continuous scores, each one for a different drug combination, ranging from negative and more synergistic values, to positive and more antagonistic values), validated against the computationally derived set of synergistic drug combinations for the AGS cell line (*gold standard*). This is a classical binary classification problem, where an imaginary threshold scans the range of Drabme’s predicted synergy scores to derive various performance scores. Specifically, each threshold demarcates the synergistic drug combinations from the antagonistic ones, based on the output prediction scores. By comparing these assessments with what we consider as the actual truth, i.e. the *gold standard* synergy set, a confusion matrix can be constructed and several threshold-specific measures calculated (Figure 6). By accounting for every such threshold, ranging from a very strict (every prediction is declared as antagonistic) to a very relaxed one (every prediction is declared as synergistic), we can visualize Drabme’s classifier performance with the Receiver Operating Characteristic (ROC) [118] and Precision Recall (PR) [119] curves, which are both well-known diagnostic tools for binary classification problems. In particular, they allow for a more broad, threshold-agnostic performance

⁷See the DrugLogics software documentation [here](#)

measurement, which is the Area Under the Curve (AUC). An AUC score of 1 is considered as the perfect classification, which in our case would happen if the top most negative synergy scores were exactly the ones corresponding to the drug combinations that represent the gold standard set for the AGS cell line. The aforementioned diagnostic curves also enable the calculation of “optimal” cutpoints, e.g. thresholds that maximize or minimize certain criteria, an example of the latter being the distance from the point of perfect classification [120]. Based on this theoretical framework, we developed the R shiny app [121] druglogics-roc to automatically parse Drabme’s output files and produce a table [122] with the confusion matrix values per synergy threshold, along with interactive plots of the ROC (using the R package plotly [123]) and PR curves (using the R package PRROC [124]) and their respective AUC scores. This app is now part of the DrugLogics software ecosystem, facilitating the visualization of the pipeline’s performance.

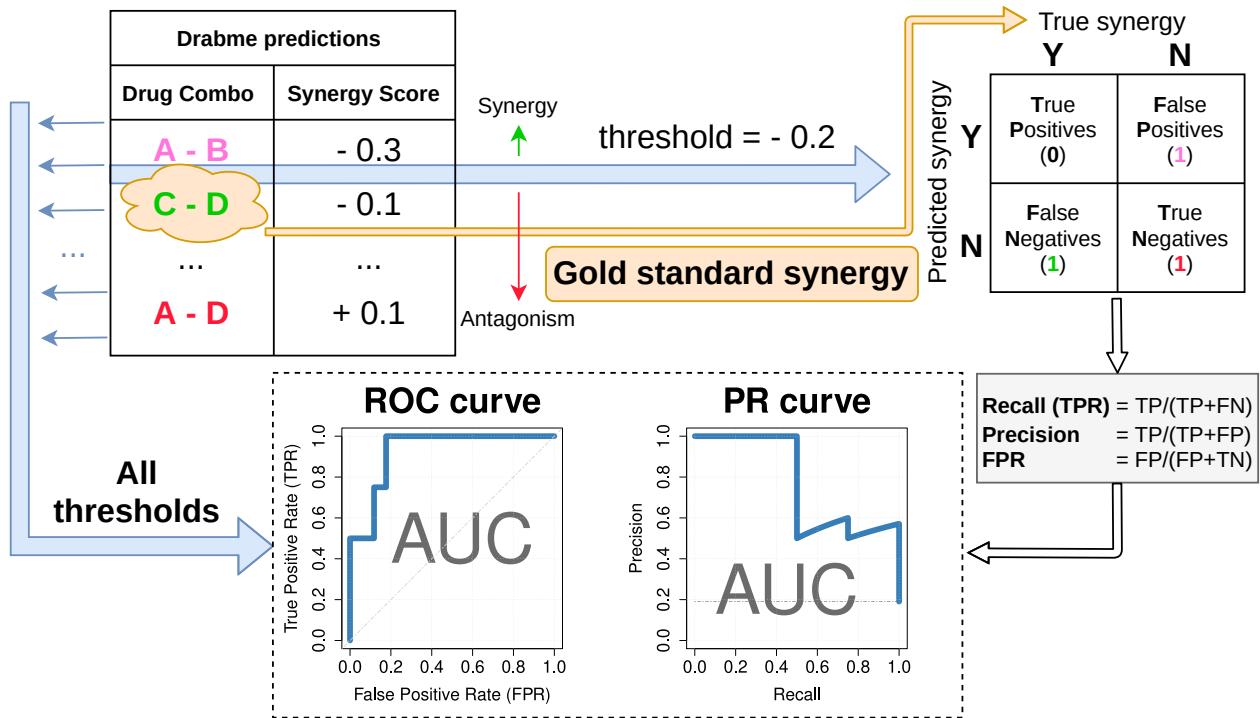


Figure 6: Performance assessment of Drabme’s drug synergy predictions. Predicted synergy scores are sorted from synergistic to antagonistic and compared to a gold standard synergy set for several possible thresholds. Each synergy threshold can be used to construct a confusion matrix, from which standard metrics are calculated, such as the number of True Positive (TP) and False Positive (FP) predictions, precision and recall, etc. Visualizing several of these metrics across all thresholds in the Receiver Operating Characteristic (ROC) and Precision Recall (PR) curves, enables the calculation of the Area Under the Curve (AUC), which is a performance score indicating how good the synergy classification method is.

The Boolean model ensembles produced by Gitsbe were a unique source for further data analyses. Since such an ensemble contains a large variety of models, all representing the same biological system, we investigated how these models differ in terms of each network node’s activity (in the respective attractor) and parameterization. Moreover, we were interested in how these types

of model differences translate to variations in prediction performance, i.e. make some models predict specific drug combinations as synergistic or not. By assigning Gitsbe models into different classes based on their individual prediction performance, we could identify nodes that were relatively more active (or inhibited) in the upper tier models compared to the lower tier models of the performance hierarchy. For example, we verified across many analyses⁸ that the ERK node of the CASCADE signaling network was particularly overexpressed in the models that predicted most of the gold standard synergies in the AGS cell line from the reference drug combination dataset. This was an interesting finding, since knowledge gathered from the scientific literature indicates conflicting measurements of ERK activity in AGS cells [90] and so our modeling results, upon further analysis, have the potential of providing useful information related to the studied biological system (this particular result is also shown using a different methodology in [Paper 3](#)). In addition to the activity-based analyses, we also explored differences with respect to the Boolean model parameterization, i.e. if higher performance models tend to have specific logical operators (or not) in some equations. This also motivated us to study how the diversity in particular Boolean rule assignments in the different model classes translates to the respective target nodes' activity (the link between node parameterization and activity was further investigated in [Paper 5](#)). To enable all the aforementioned investigations, we began writing functions in several scripts while getting familiar with the world of professional software development in R [125]. In the end, we spent considerable effort to organize all these functions into a single, *clean*, modularized and tested codebase, with the purpose to fill in a niche for data analysis-oriented software that performs auxiliary automated analyses on Boolean model datasets. The result was the creation of the emba R package and its addition to the DrugLogics software ecosystem ([Paper 4](#)).

⁸See repository gitsbe-model-analysis [here](#)

Get the right rules

In the previous chapter we described in detail all the different requirements that a software needs to satisfy so as to deliver on its promise to *do what it was made to do*. In other words, it's entirely the software developer's responsibility to make sure that the underlying algorithms are programmed correctly, the model equations and their solutions are right and that in general, the software works as expected. In the case of our modeling pipeline, this translates for example to the precise and error-free implementation of the genetic algorithm as well as taking the extra effort to test and ensure that the Boolean models take on the desired parameterization and their attractors are correctly calculated. The aforementioned procedure is semantically synonymous to *verification*: the software works in a manner that directly reflects the underlying theories and modeling assumptions. This is part of what makes the simulation results trustworthy and actionable, in the sense that they can be used to provide solutions to real-world problems, making the respective models valuable for diverse applications, both in industrial and clinical contexts.

It's an entirely different matter if the solutions that the software was made to produce, (e.g. the simulation results in the case of modeling software), are pragmatic. What good are models if their outputs do not agree with real observations, no matter how skillfully the developer put the theoretical ideas in software code? All models are wrong since they are approximate representations of reality, but it's important to be able to pinpoint exactly where this wrongness originates and what it pertains to. If a biological model for example cannot reproduce basic observations about the phenotype of the system it simulates (via its respective software), then that model is not useful and it probably needs further refinement. That leads to the question of what makes a model able to better match experimental data and become a more faithful approximation of reality. In other words, what aspect of the model needs refinement? The first step towards answering this question is understanding that a model is more than just the code. A model, as explained in a previous chapter, is its mathematical rules (also referred to as model parameterization). So, extra care should be given to make sure we have the right equations in our models. This crucial next step is known as the *validation* of the modeling software and its basic assumption is that more "right" equations result in better fit to observations, which lead to better models (their behavior corresponds more accurately to reality) and subsequently, better simulation results (predictions about the studied system).

So for every modeling application and subsequent software, not only do we need to have the rules right (*verification*), but it is of equal or maybe even more significance to have the right rules (*validation*) [126]. In that aspect, the parameterization of the Boolean models in our software, i.e. the choice of logical equations that define the regulatory activity of every target in the underlying cancer signaling network, stands as one of the most important parts of the ensuing modeling process. Since parameterization is so important, how can we find which are the "right" rules or similarly, how can we distinguish between those that are right vs those that aren't (or are less so), so that

we can choose the former for our models? Practically, the way that researchers in the logical modeling community have dealt with this problem is by establishing standard logical equation forms to represent regulatory interactions upon a signaling target [127]. Based on such a foundation for the initial construction of the logical rules, the next step is to tweak them properly, by changing logical operators and removing/adding variables (regulators), so that a better match with experimental observations can be achieved. This process of calibrating the rules to fit the respective data can be either the result of manual curation [90,128,129], or the outcome of automated computational search for optimal logical equations [130,131] (for similar efforts in this thesis see [Paper 3](#)). In addition, several other methods are used to convert various input sources to the appropriate Boolean rules, e.g. by translating molecular interaction maps directly to Boolean models [132] or user-provided text to suitable logical equations via web-based tools [133].

Be it the modelers or the computers that refine and produce the final Boolean equations in the respective models, we reasoned that a proper framework to characterize the Boolean functions that constitute the rules in these models, is currently missing. The main idea is that, since the mathematical rules are the heart of modeling and we need to have the right rules (or as best as possible) for validating and further refining our models, a proper toolkit is needed to differentiate and choose between the possible parameterization options. The plethora of potential equations that can be just “right” are a direct consequence of the fact that the number of possible parameterizations increases dramatically with the number of regulators [134]. In addition, a large number of Boolean equations may fit equally well the experimental observations or it might even be the case that the data are not enough to uniquely define the model equations [135]. Either way, fitting the model to match the expected outputs is just one side of the validation process. We need to go deeper than that though if we are to reach our goal of finding the right rules for our models. To establish a practical framework assisting in the choice of a particular logical model parameterization, we need to search for ways to expand our knowledge of the rules and gain insights into these from different perspectives. Following this logic, there are function properties that Boolean mathematics research has thoroughly studied and which, when brought to the context of modeling, can be beneficial for both modelers and software applications that specify or calibrate the rules of a logical model [134,136,137]. For example, such properties could be used to investigate if the equations contradict the structure of the underlying regulatory topology (PKN) or that biologically important regulators manifest in the equations as proportionally influential to the respective Boolean output. Our efforts constitute a first attempt to compile such a list of Boolean function metrics in a unified framework that could be used to refine the search for the optimal rules to use ([Paper 5](#)). The analyses in that paper also show the differences between varying Boolean parameterizations in terms of expected output behavior and how this information, when known *a priori* (based on precise mathematical formalization and subsequent calculation), can assist in the choice of better rules for the considered models.

While establishing a framework to help in the identification of the right rules for a Boolean model, we leveraged the benefits of variation in model parameterization (Figure 7). Exploring the effects of variation in the values of model parameters, enables the identification of variables which have the largest influence in the behavior of the studied systems and can provide mechanistic insights to explain several phenomena [75]. For example, in Eduati et al. (2017) [86], the authors study how logical model parameters are related with cellular sensitivity to anti-cancer drugs. Simply put, by tweaking a biologically meaningful parameter in their model (the responsiveness of the GSK3 signaling node), you could explain why a drug combination involving a MEK and a GSK3 inhibitor is sensitive in some cancer cell lines and resistant in others. This computational finding was supported by experimental evidence, thereby providing a proof-of-concept in how the investigation of model parameterization can suggest new mechanisms for the manifestation of particular drug synergies. On a more theoretical front, in Abou-Jaoudé et al. (2019) [138], the authors formally describe the concept of logical bifurcation diagrams, a framework to assess how changes in the logical parameters result in the change of the dynamics of logical models. This methodology was used to display the attractors of the simple p53-Mdm2 signaling network as a function of the degradation rate of ubiquitin ligase Mdm2 in the nucleus, allowing for a more concise characterization of its main dynamical properties [139].

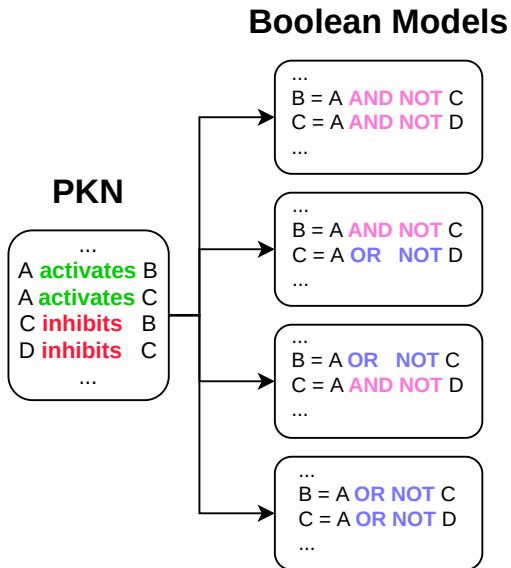


Figure 7: Exploring Boolean model parameterization with the `abmlog` software. Using as input causal regulatory knowledge, all possible Boolean models that conform to a standardized logical equation form [127] and its most basic variation are produced. Target nodes B and C have two regulators each, and their respective Boolean equations can be formalized in two ways, producing a total of four possible models for further analysis.

Such approaches inspired us to explore the space of Boolean model parameterization, pertaining to the equations used to construct and mutate the models in the genetic algorithm of Gitsbe ([Paper 3](#)). We made a Java software package that can produce either a random sample or all possible Boolean models, based on a standardized equation form [127] and it's most basic variation (Figure 7). The `abmlog` package is now also part of the DrugLogics software ecosystem and was used in the analyses of [Paper 5](#) to show how the output behavior of the two alternative parameterization options of the Gitsbe models varies based on the number of regulators in the respective equations and the ratio of positive to negative regulators (as these were defined in the original PKN). Moreover, a large number of CASCADE-based Boolean models were produced using the `abmlog` package to explore parameterization variation. Exploiting the dimension reduction and visualization method UMAP [140], we constructed several Boolean model maps and were able to visualize model differences such as fitness to training data and prediction performance.⁹ Further analysis identified important nodes which drive the change of dynamics (number of attractors in the CASCADE Boolean models) and whose parameterization could be used to visually separate the UMAP embedded models in distinct clusters.

⁹See repository `bool-param-maps` [here](#)

Links to software, documentation and data analyses

GitHub organizations

- UniBioDicts: <https://github.com/UniBioDicts/>
- VSM: <https://github.com/vsm>
- DrugLogics: <https://github.com/druglogics>
- PSICQUIC: <https://github.com/PSICQUIC>

Documentation

- DrugLogics software: <https://druglogics.github.io/druglogics-doc/>
- VSM technology and related projects: <https://vsm.github.io/>
- PSICQUIC: <https://psicquic.github.io/>

DrugLogics software modules

- [gitsbe](#): A Java module that defines Boolean models compliant with observed behavior (e.g. steady state or perturbation data) using an automated, model parameterization genetic algorithm
- [drabme](#): A Java module that performs a drug perturbation response analysis to the Boolean model ensembles generated by Gitsbe
- [druglogics-synergy](#): A Java module to execute serially Gitsbe and then Drabme
- [abmlog](#): A Java-based generator of all possible logical models with AND/OR-NOT link operators in their respective Boolean equations (Figure 7)
- [druglogics-roc](#): R Shiny web app to visualize the ROC and PR prediction performance of Drabme's ensemble-wise predictions (Figure 6)
- [emba](#): R package for analysis and visualization of biomarkers in Boolean model ensembles [3]

R community packages

- [usefun](#): various useful R functions
- [rtemp](#)s: templates for reproducible data analyses with R [141]

Miscellaneous data analyses and repositories

All the following repositories have been authored exclusively by myself. Each repository has a README.md file with a brief description of the analysis and a link to online documentation and results (in the form of R Markdown documents or R GitBooks).

- [CASCADE](#): repository of the different versions of the CAncer Signaling CAusality DatabasE developed by the DrugLogics group
- [ags-paper](#): simulation results and data analyses related to [Paper 3](#)
- [sintef-obs-synergies](#): synergy assessment of the Flobak et al. (2019) [115] drug combination dataset using rbbt [142] and the CImbinator tool [116]
- [brf-bias](#): data analyses related to the study of truth density bias in standardized Boolean regulatory functions for [Paper 5](#)
- [gitsbe-model-analysis](#): several analyses using Boolean model ensemble datasets generated via Gitsbe and the emba R package to analyze them
- [bool-param-maps](#): visualization of model parameterization and node importance using UMAP and random forests on the CASCADE 1.0 Boolean model dataset generated by abmlog

References

1. Zobolas, J., Touré, V., Kuiper, M., & Vercruyse, S. (2020). UniBioDicts: Unified access to Biological Dictionaries. *Bioinformatics*, 37(1), 143–144. <https://doi.org/10.1093/bioinformatics/btaa1065>
2. Zobolas, J., Kim, J.-D., Kuiper, M., & Vercruyse, S. (2020). *Linking PubDictionaries with UniBioDicts to support Community Curation*. BioHackrXiv. <https://doi.org/10.37044/osf.io/gzfa8>
3. Zobolas, J., Kuiper, M., & Flobak, Å. (2020). emba: R package for analysis and visualization of biomarkers in boolean model ensembles. *Journal of Open Source Software*, 5(53), 2583. <https://doi.org/10.21105/joss.02583>
4. Zobolas, J., Monteiro, P. T., Kuiper, M., & Flobak, Å. (2021). *Boolean function metrics can assist modelers to check and choose logical rules*. <http://arxiv.org/abs/2104.01279>
5. Vercruyse, S., Zobolas, J., Touré, V., Andersen, M. K., & Kuiper, M. (2020). VSM-box: general-purpose interface for biocuration and knowledge representation. *Preprints*. <https://doi.org/10.20944/preprints202007.0557.v1>
6. Touré, V., Zobolas, J., Kuiper, M., & Vercruyse, S. (2021). CausalBuilder: bringing the MI2CAST causal interaction annotation standard to the curator. *Database*. <https://doi.org/10.1093/database/baaa107>
7. Perfetto, L., Acencio, M. L., Bradley, G., Cesareni, G., Del Toro, N., Fazekas, D., Hermjakob, H., Korcsmaros, T., Kuiper, M., Lægreid, A., Lo Surdo, P., Lovering, R. C., Orchard, S., Porras, P., Thomas, P. D., Touré, V., Zobolas, J., & Licata, L. (2019). CausalTAB: the PSI-MITAB 2.8 updated format for signalling data representation and dissemination. *Bioinformatics*. <https://doi.org/10.1093/bioinformatics/btz132>
8. Abbate, J. (2000). *Inventing the internet*. MIT press.
9. Naughton, J. (2016). The evolution of the Internet: from military experiment to General Purpose Technology. *Journal of Cyber Policy*, 1(1), 5–28. <https://doi.org/10.1080/23738871.2016.1157619>
10. Polasky, S., Kling, C. L., Levin, S. A., Carpenter, S. R., Daily, G. C., Ehrlich, P. R., Heal, G. M., & Lubchenco, J. (2019). Role of economics in analyzing the environment and sustainable

- development. *Proceedings of the National Academy of Sciences of the United States of America*, 116(12), 5233–5238. <https://doi.org/10.1073/pnas.1901616116>
11. Roser, M., Ortiz-Ospina, E., & Ritchie, H. (2013). *Life Expectancy*. <https://ourworldindata.org/life-expectancy> (15 May 2021, date last accessed).
 12. Jinek, M., Chylinski, K., Fonfara, I., Hauer, M., Doudna, J. A., & Charpentier, E. (2012). A programmable dual-RNA–guided DNA endonuclease in adaptive bacterial immunity. *Science*, 337(6096), 816–821.
 13. Zhu, H., Li, C., & Gao, C. (2020). Applications of CRISPR–Cas in agriculture and plant biotechnology. *Nature Reviews Molecular Cell Biology*, 21(11), 661–677. <https://doi.org/10.1038/s41580-020-00288-9>
 14. Bailey, R., & Tupy, M. L. (2020). *Ten Global Trends Every Smart Person Should Know: And Many Others You Will Find Interesting*. Cato Institute.
 15. Gibbons, M. (1999). Science’s new social contract with society. *Nature*, 402, C81–C84. <https://doi.org/10.1038/35011576>
 16. *Healthcare Access and Quality Index*. (2015). <https://ourworldindata.org/grapher/healthcare-access-and-quality-index> (15 May 2021, date last accessed).
 17. Apweiler, R., Beissbarth, T., Berthold, M. R., Blüthgen, N., Burmeister, Y., Dammann, O., Deutsch, A., Feuerhake, F., Franke, A., Hasenauer, J., Hoffmann, S., Höfer, T., Jansen, P. L., Kaderali, L., Klingmüller, U., Koch, I., Kohlbacher, O., Kuepfer, L., Lammert, F., ... Wolkenhauer, O. (2018). Whither systems medicine? *Experimental & Molecular Medicine*, 50(3), e453. <https://doi.org/10.1038/emm.2017.290>
 18. Trudeau, R. J. (1976). *Dots and lines*. Kent State University Press.
 19. Erdos, P., & Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1), 17–60.
 20. Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439), 509–512. <https://doi.org/10.1126/science.286.5439.509>
 21. Barabási, A. L. (2013). Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987). <https://doi.org/10.1098/rsta.2012.0375>
 22. Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5), 1–9. <http://igraph.org>
 23. Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. *Proceedings of the International AAAI Conference on Web and Social Media*.
 24. Mrvar, A., & Batagelj, V. (2016). Analysis and visualization of large networks with program

- package Pajek. *Complex Adaptive Systems Modeling*, 4(1), 1–8. <https://doi.org/10.1186/s40294-016-0017-8>
25. Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., & Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11), 2498–2504.
 26. Dahlquist, K. D., Salomonis, N., Vranizan, K., Lawlor, S. C., & Conklin, B. R. (2002). GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nature Genetics*, 31(1), 19–20. <https://doi.org/10.1038/ng0502-19>
 27. Breitkreutz, B.-J., Stark, C., & Tyers, M. (2003). Osprey: a network visualization system. *Genome Biology*, 4(3), R22. <https://doi.org/10.1186/gb-2003-4-3-r22>
 28. Funahashi, A., Morohashi, M., Kitano, H., & Tanimura, N. (2003). CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO*, 1(5), 159–162. [https://doi.org/10.1016/s1478-5382\(03\)02370-9](https://doi.org/10.1016/s1478-5382(03)02370-9)
 29. Sidiropoulos, K., Viteri, G., Sevilla, C., Jupe, S., Webber, M., Orlic-Milacic, M., Jassal, B., May, B., Shamovsky, V., Duenas, C., Rothfels, K., Matthews, L., Song, H., Stein, L., Haw, R., D'Eustachio, P., Ping, P., Hermjakob, H., & Fabregat, A. (2017). Reactome enhanced pathway visualization. *Bioinformatics*, 33(21), 3461–3467. <https://doi.org/10.1093/bioinformatics/btx441>
 30. Smith, M. A., Shneiderman, B., Milic-Frayling, N., Mendes Rodrigues, E., Barash, V., Dunne, C., Capone, T., Perer, A., & Gleave, E. (2009). Analyzing (social media) networks with NodeXL. *Proceedings of the Fourth International Conference on Communities and Technologies - c&t '09*, 255. <https://doi.org/10.1145/1556460.1556497>
 31. Kalamaras, D. (2014). *Social Networks Visualizer (SocNetV): Social network analysis and visualization software*. <http://socnetv.org>
 32. Chatterjee, T., Albert, R., Thapliyal, S., Azarhooshang, N., & DasGupta, B. (2021). Detecting network anomalies using Forman–Ricci curvature and a case study for human brain networks. *Scientific Reports*, 11(1), 8121. <https://doi.org/10.1038/s41598-021-87587-z>
 33. Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., & Bhattacharjee, B. (2007). Measurement and analysis of online social networks. *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, 29–42. <https://doi.org/10.1145/1298306.1298311>
 34. Maheshwari, P., & Albert, R. (2020). Network model and analysis of the spread of Covid-19 with social distancing. *Applied Network Science*, 5(1), 1–13. <https://doi.org/10.1007/s41109-020-00344-5>
 35. Barabási, A.-L., Gulbahce, N., & Loscalzo, J. (2011). Network medicine: a network-based approach to human disease. *Nature Reviews. Genetics*, 12(1), 56–68. <https://doi.org/10.1038/nrg2918>
 36. Vercruyse, S. (2019). *VSM Pages*. <https://vsm.github.io/vsm-pages/intro> (15 May 2021, date

- last accessed).
37. Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Hill, D. P., Kania, R., Schaeffer, M., St Pierre, S., Twigger, S., White, O., & Yon Rhee, S. (2008). Big data: The future of biocuration. *Nature*, 455(7209), 47–50. <https://doi.org/10.1038/455047a>
 38. Ammari, M., Chatr Aryamontri, A., Attrill, H., Bairoch, A., Berardini, T., Blake, J., Chen, Q., Collado, J., Dauga, D., Dudley, J. T., Engel, S., Erill, I., Fey, P., Gibson, R., Hermjakob, H., Holliday, G., Howe, D., Hunter, C., Landsman, D., ... Zhang, Z. (2018). Biocuration: Distilling data into knowledge. *PLOS Biology*, 16(4), e2002846. <https://doi.org/10.1371/journal.pbio.2002846>
 39. Jenssen, T.-K., Lægreid, A., Komorowski, J., & Hovig, E. (2001). A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1), 21–28. <https://doi.org/10.1038/ng0501-21>
 40. *IntAct editor*. (2007). <https://github.com/EBI-IntAct/intact-editor> (15 May 2021, date last accessed).
 41. Rutherford, K. M., Harris, M. A., Lock, A., Oliver, S. G., & Wood, V. (2014). Canto: an online tool for community literature curation. *Bioinformatics*, 30(12), 1791–1792. <https://doi.org/10.1093/bioinformatics/btu103>
 42. *Canto Documentation*. (2014). <https://curation.pombe.org/pombe/docs/index/>, (15 May 2021, date last accessed).
 43. Kuperstein, I., Cohen, D. P. A., Pook, S., Viara, E., Calzone, L., Barillot, E., & Zinovyev, A. (2013). NaviCell: A web-based environment for navigation, curation and maintenance of large molecular interaction maps. *BMC Systems Biology*, 7(1), 100. <https://doi.org/10.1186/1752-0509-7-100>
 44. Gawron, P., Ostaszewski, M., Satagopam, V., Gebel, S., Mazein, A., Kuzma, M., Zorzan, S., McGee, F., Otjacques, B., Balling, R., & Schneider, R. (2016). MINERVA—A platform for visualization and curation of molecular interaction networks. *Npj Systems Biology and Applications*, 2(1), 1–6. <https://doi.org/10.1038/npjbsa.2016.20>
 45. Novère, N. L., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M. I., Wimalaratne, S. M., Bergman, F. T., Gauges, R., Ghazal, P., Kawaji, H., Li, L., Matsuoka, Y., Villéger, A., Boyd, S. E., Calzone, L., ... Kitano, H. (2009). The Systems Biology Graphical Notation. *Nature Biotechnology*, 27(8), 735–741. <https://doi.org/10.1038/nbt.1558>
 46. Vercruyse, S., & Kuiper, M. (2020). Intuitive representation of computable knowledge. *Preprints*. <https://doi.org/10.20944/preprints202007.0486.v2>
 47. Touré, V., Vercruyse, S., Acencio, M. L., Lovering, R., Orchard, S., Bradley, G., Casals-Casas, C., Chaouiya, C., Del-Toro, N., Flobak, Å., Gaudet, P., Hermjakob, H., Licata, L., Lægreid, A., Mungall, C., Niknejad, A., Panni, S., Perfetto, L., Porras, P., ... Kuiper, M.

- (2020). The Minimum Information about a Molecular Interaction Causal Statement (MI2CAST). *Bioinformatics*. <https://doi.org/10.1093/bioinformatics/btaa622>
48. Vasundra, T. (2020). *MI2CAST Documentation*. https://github.com/MI2CAST/MI2CAST/blob/master/docs/MI2CAST_guideline.md (15 May 2021, date last accessed).
49. The UniProt Consortium. (2019). UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1), D506–D515. <https://doi.org/10.1093/nar/gky1049>
50. Meldal, B. H. M., Bye-A-Jee, H., Gajdoš, L., Hammerová, Z., Horáčková, A., Melicher, F., Perfetto, L., Pokorný, D., Lopez, M. R., Türková, A., Wong, E. D., Xie, Z., Casanova, E. B., Del-Toro, N., Koch, M., Porras, P., Hermjakob, H., & Orchard, S. (2019). Complex Portal 2018: extended content and enhanced visualization tools for macromolecular complexes. *Nucleic Acids Research*, 47(D1), D550–D558. <https://doi.org/10.1093/nar/gky1001>
51. The RNAcentral Consortium. (2018). RNACentral: a hub of information for non-coding RNA sequences. *Nucleic Acids Research*, 47(D1), D1250–D1251. <https://doi.org/10.1093/nar/gky1206>
52. Whetzel, P. L., Noy, N. F., Shah, N. H., Alexander, P. R., Nyulas, C., Tudorache, T., & Musen, M. A. (2011). BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Research*, 39(Web Server issue), W541–5. <https://doi.org/10.1093/nar/gkr469>
53. Madeira, F., Park, Y. mi, Lee, J., Buso, N., Gur, T., Madhusoodanan, N., Basutkar, P., Tivey, A. R. N., Potter, S. C., Finn, R. D., & Lopez, R. (2019). The EMBL-EBI search and sequence analysis tools APIs in 2019. *Nucleic Acids Research*, 47(W1), W636–W641. <https://doi.org/10.1093/nar/gkz268>
54. Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J. W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., ... Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1), 1–9. <https://doi.org/10.1038/sdata.2016.18>
55. Hartmann, N. B., Hüffer, T., Thompson, R. C., Hassellöv, M., Verschoor, A., Daugaard, A. E., Rist, S., Karlsson, T., Brennholt, N., Cole, M., Herrling, M. P., Hess, M. C., Ivleva, N. P., Lusher, A. L., & Wagner, M. (2019). Are We Speaking the Same Language? Recommendations for a Definition and Categorization Framework for Plastic Debris. *Environmental Science and Technology*, 53(3), 1039–1047. <https://doi.org/10.1021/acs.est.8b05297>
56. Kim, J.-D., Wang, Y., Fujiwara, T., Okuda, S., Callahan, T. J., & Cohen, K. B. (2019). Open Agile text mining for bioinformatics: the PubAnnotation ecosystem. *Bioinformatics*, 35(21), 4372–4380. <https://doi.org/10.1093/bioinformatics/btz227>
57. Hermjakob, H., Montecchi-Palazzi, L., Bader, G., Wojcik, J., Salwinski, L., Ceol, A., Moore, S.,

- Orchard, S., Sarkans, U., Mering, C. von, Roechert, B., Poux, S., Jung, E., Mersch, H., Kersey, P., Lappe, M., Li, Y., Zeng, R., Rana, D., ... Apweiler, R. (2004). The HUPO PSI's Molecular Interaction format—a community standard for the representation of protein interaction data. *Nature Biotechnology*, 22(2), 177–183. <https://doi.org/10.1038/nbt926>
58. Sivade, M., Alonso-López, D., Ammari, M., Bradley, G., Campbell, N. H., Ceol, A., Cesareni, G., Combe, C., De Las Rivas, J., Del-Toro, N., Heimbach, J., Hermjakob, H., Jurisica, I., Koch, M., Licata, L., Lovering, R. C., Lynn, D. J., Meldal, B. H. M., Micklem, G., ... Orchard, S. (2018). Encompassing new use cases - level 3.0 of the HUPO-PSI format for molecular interactions. *BMC Bioinformatics*, 19(1), 134. <https://doi.org/10.1186/s12859-018-2118-1>
59. Kerrien, S., Orchard, S., Montecchi-Palazzi, L., Aranda, B., Quinn, A. F., Vinod, N., Bader, G. D., Xenarios, I., Wojcik, J., Sherman, D., Tyers, M., Salama, J. J., Moore, S., Ceol, A., Chatr-aryamontri, A., Oesterheld, M., Stümpflen, V., Salwinski, L., Nerothin, J., ... Hermjakob, H. (2007). Broadening the horizon – level 2.5 of the HUPO-PSI format for molecular interactions. *BMC Biology*, 5(1), 44. <https://doi.org/10.1186/1741-7007-5-44>
60. Licata, L., Lo Surdo, P., Iannuccelli, M., Palma, A., Micarelli, E., Perfetto, L., Peluso, D., Calderone, A., Castagnoli, L., & Cesareni, G. (2019). SIGNOR 2.0, the SIGnaling Network Open Resource 2.0: 2019 update. *Nucleic Acids Research*. <https://doi.org/10.1093/nar/gkz949>
61. Aranda, B., Blankenburg, H., Kerrien, S., Brinkman, F. S. L., Ceol, A., Chautard, E., Dana, J. M., De Las Rivas, J., Dumousseau, M., Galeota, E., Gaulton, A., Goll, J., Hancock, R. E. W., Isserlin, R., Jimenez, R. C., Kerssemakers, J., Khadake, J., Lynn, D. J., Michaut, M., ... Hermjakob, H. (2011). PSICQUIC and PSISCORE: accessing and scoring molecular interactions. *Nature Methods*, 8(7), 528–529. <https://doi.org/10.1038/nmeth.1637>
62. *MITAB 2.8 Documentation*. (2018). <http://psicquic.github.io/MITAB28Format.html>, (15 May 2021, date last accessed).
63. *Molecular Interactions Community*. (2015). <https://github.com/MICommunity>, (15 May 2021, date last accessed).
64. *Software rot (Wikipedia entry)*. (2014). https://en.wikipedia.org/wiki/Software_rot, (15 May 2021, date last accessed).
65. Sivade, M., Koch, M., Shrivastava, A., Alonso-López, D., De Las Rivas, J., Del-Toro, N., Combe, C. W., Meldal, B. H. M., Heimbach, J., Rappsilber, J., Sullivan, J., Yehudi, Y., & Orchard, S. (2018). JAMI: a Java library for molecular interactions and data interoperability. *BMC Bioinformatics*, 19(1), 133. <https://doi.org/10.1186/s12859-018-2119-0>
66. *GREEKC Hinxton Workshop*. (2018). <https://www.greekc.org/activity/hinxton-workshop/>, (15 May 2021, date last accessed).
67. *ELIXIR BioHackathon Paris*. (2018). <https://2018.biohackathon-europe.org/>, (15 May 2021, date last accessed).

68. Del-Toro, N., Zobolas, J., & Touré, V. (2019). *Signor Dataset in CausalTAB (PSICQUIC Dev Server)*. http://wwwdev.ebi.ac.uk/Tools/webservices/psicquic/causality/webservices/current/search/query/*?firstResult=0&maxResults=10&format=tab28, (15 May 2021, date last accessed).
69. *PSICQUIC View*. (2011). <http://www.ebi.ac.uk/Tools/webservices/psicquic/view/main.xhtml>, (15 May 2021, date last accessed).
70. *PSICQUIC Universal Client (Cytoscape app)*. (2012). https://apps.cytoscape.org/apps/psicquic_universalclient, (15 May 2021, date last accessed).
71. Millán, P. P. (2013). Visualization and analysis of biological networks. *Methods in Molecular Biology*, 1021, 63–88. https://doi.org/10.1007/978-1-62703-450-0_4
72. Shannon, P. (2020). *PSICQUIC: Proteomics Standard Initiative Common QUery InterfaCe. R package version 1.28.0*. <https://doi.org/10.18129/B9.bioc.PSICQUIC>
73. Kleshchevnikov, V. (2021). *PItools: Protein interaction data tools. R package*. GitHub. <https://github.com/vitkl/PItools>
74. Wang, R.-S., Saadatpour, A., & Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5), 55001. <https://doi.org/10.1088/1478-3975/9/5/055001>
75. Aldridge, B. B., Burke, J. M., Lauffenburger, D. A., & Sorger, P. K. (2006). Physicochemical modelling of cell signalling pathways. *Nature Cell Biology*, 8(11), 1195–1203. <https://doi.org/10.1038/ncb1497>
76. ElKalaawy, N., & Wassal, A. (2015). Methodologies for the modeling and simulation of biochemical networks, illustrated for signal transduction pathways: A primer. *Biosystems*, 129, 1–18. <https://doi.org/10.1016/J.BIOSYSTEMS.2015.01.008>
77. Groen, D., Arabnejad, H., Jancauskas, V., Edeling, W. N., Jansson, F., Richardson, R. A., Lakhili, J., Veen, L., Bosak, B., Kopta, P., Wright, D. W., Monnier, N., Karlshoefer, P., Suleimenova, D., Sinclair, R., Vassaux, M., Nikishova, A., Bieniek, M., Luk, O. O., ... Coveney, P. V. (2021). VECMAtk: a scalable verification, validation and uncertainty quantification toolkit for scientific simulations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2197). <https://doi.org/10.1098/rsta.2020.0221>
78. Schwab, J. D., Kühlwein, S. D., Ikonomi, N., Kühl, M., & Kestler, H. A. (2020). Concepts in Boolean network modeling: What do they all mean? *Computational and Structural Biotechnology Journal*, 18, 571–582. <https://doi.org/10.1016/j.csbj.2020.03.001>
79. Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3), 437–467. [https://doi.org/10.1016/0022-5193\(69\)90015-0](https://doi.org/10.1016/0022-5193(69)90015-0)
80. Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA.
81. Faure, A., Naldi, A., Chaouiya, C., & Thieffry, D. (2006). Dynamical analysis of a generic

- Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14), e124–e131. <https://doi.org/10.1093/bioinformatics/btl210>
82. Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., & Chaouiya, C. (2016). Logical Modeling and Dynamical Analysis of Cellular Networks. *Frontiers in Genetics*, 7, 94. <https://doi.org/10.3389/fgene.2016.00094>
83. Naldi, A., Hernandez, C., Levy, N., Stoll, G., Monteiro, P. T., Chaouiya, C., Helikar, T., Zinovyev, A., Calzone, L., Cohen-Boulakia, S., Thieffry, D., & Paulevé, L. (2018). The CoLoMoTo Interactive Notebook: Accessible and Reproducible Computational Analyses for Qualitative Biological Networks. *Frontiers in Physiology*, 9, 680. <https://doi.org/10.3389/fphys.2018.00680>
84. Hood, L., & Friend, S. H. (2011). Predictive, personalized, preventive, participatory (P4) cancer medicine. *Nature Reviews Clinical Oncology*, 8(3), 184–187. <https://doi.org/10.1038/nrclinonc.2010.227>
85. Béal, J., Montagud, A., Traynard, P., Barillot, E., & Calzone, L. (2018). Personalization of Logical Models With Multi-Omics Data Allows Clinical Stratification of Patients. *Frontiers in Physiology*, 9, 1965. <https://doi.org/10.3389/fphys.2018.01965>
86. Eduati, F., Doldà-Martelli, V., Klinger, B., Cokelaer, T., Sieber, A., Kogera, F., Dorel, M., Garnett, M. J., Blüthgen, N., & Saez-Rodriguez, J. (2017). Drug Resistance Mechanisms in Colorectal Cancer Dissected with Cell Type-Specific Dynamic Logic Models. *Cancer Research*, 77(12), 3364–3375. <https://doi.org/10.1158/0008-5472.CAN-17-0078>
87. Béal, J., Pantolini, L., Noël, V., Barillot, E., & Calzone, L. (2021). Personalized logical models to investigate cancer response to BRAF treatments in melanomas and colorectal cancers. *PLOS Computational Biology*, 17(1), e1007900. <https://doi.org/10.1371/journal.pcbi.1007900>
88. Tognetti, M., Gabor, A., Yang, M., Cappelletti, V., Windhager, J., Rueda, O. M., Charmpi, K., Esmaeilishirazifard, E., Bruna, A., Souza, N. de, Caldas, C., Beyer, A., Picotti, P., Saez-Rodriguez, J., & Bodenmiller, B. (2021). Deciphering the signaling network of breast cancer improves drug sensitivity prediction. *Cell Systems*, 12. <https://doi.org/10.1016/j.cels.2021.04.002>
89. Saadatpour, A., Wang, R.-S., Liao, A., Liu, X., Loughran, T. P., Albert, I., & Albert, R. (2011). Dynamical and Structural Analysis of a T Cell Survival Network Identifies Novel Candidate Therapeutic Targets for Large Granular Lymphocyte Leukemia. *PLoS Computational Biology*, 7(11), e1002267. <https://doi.org/10.1371/journal.pcbi.1002267>
90. Flobak, Å., Baudot, A., Remy, E., Thommesen, L., Thieffry, D., Kuiper, M., & Lægreid, A. (2015). Discovery of Drug Synergies in Gastric Cancer Cells Predicted by Logical Modeling. *PLOS Computational Biology*, 11(8). <https://doi.org/10.1371/journal.pcbi.1004426>
91. Eduati, F., Jaaks, P., Wappler, J., Cramer, T., Merten, C. A., Garnett, M. J., & Saez-Rodriguez, J. (2020). Patient-specific logic models of signaling pathways from screenings on cancer biopsies to prioritize personalized combination therapies. *Molecular Systems Biology*, 16(2).

- <https://doi.org/10.15252/msb.20188664>
92. Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P., & Wilson, P. (2014). Best Practices for Scientific Computing. *PLoS Biology*, 12(1), e1001745. <https://doi.org/10.1371/journal.pbio.1001745>
 93. Coveney, P. V., & Highfield, R. R. (2021). When we can trust computers (and when we can't). *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 379(2197), 20200067. <https://doi.org/10.1098/rsta.2020.0067>
 94. Barnes, N. (2010). Publish your computer code: it is good enough. *Nature*, 467(7317), 753–753. <https://doi.org/10.1038/467753a>
 95. Prlić, A., & Procter, J. B. (2012). Ten Simple Rules for the Open Development of Scientific Software. *PLoS Computational Biology*, 8(12), e1002802. <https://doi.org/10.1371/journal.pcbi.1002802>
 96. Karimzadeh, M., & Hoffman, M. M. (2018). Top considerations for creating bioinformatics software documentation. *Briefings in Bioinformatics*, 19(4), 693–699. <https://doi.org/10.1093/bib/bbw134>
 97. List, M., Ebert, P., & Albrecht, F. (2017). Ten Simple Rules for Developing Usable Software in Computational Biology. *PLoS Computational Biology*, 13(1), e1005265. <https://doi.org/10.1371/journal.pcbi.1005265>
 98. Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLoS Computational Biology*, 9(10), e1003285. <https://doi.org/10.1371/journal.pcbi.1003285>
 99. Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Pearson Education.
 100. Gaddum, J. H. (1940). *Pharmacology*. Oxford University Press, London.
 101. Bliss, C. I. (1939). The Toxicity of Poisons Applied Jointly. *Annals of Applied Biology*, 26(3), 585–615. <https://doi.org/10.1111/j.1744-7348.1939.tb06990.x>
 102. Saez-Rodriguez, J., Costello, J. C., Friend, S. H., Kellen, M. R., Mangavite, L., Meyer, P., Norman, T., & Stolovitzky, G. (2016). Crowdsourcing biomedical research: Leveraging communities as innovation engines. *Nature Reviews Genetics*, 17(8), 470–486. <https://doi.org/10.1038/nrg.2016.69>
 103. Flobak, Å. (2016). Systems Medicine: From Modeling Systems Perturbations to Predicting Drug Synergies. *PhD Thesis*. <https://ntuopen.ntnu.no/ntnu-xmlui/handle/11250/2385695>
 104. Apache Commons. (2021). <https://commons.apache.org/>, (15 May 2021, date last accessed).
 105. Apache Maven. (2021). <https://maven.apache.org/>, (15 May 2021, date last accessed).

106. Naldi, A. (2018). BioLQM: A Java Toolkit for the Manipulation and Conversion of Logical Qualitative Models of Biological Networks. *Frontiers in Physiology*, 9, 1605. <https://doi.org/10.3389/fphys.2018.01605>
107. Naldi, A., Hernandez, C., Abou-Jaoudé, W., Monteiro, P. T., Chaouiya, C., & Thieffry, D. (2018). Logical Modeling and Analysis of Cellular Regulatory Networks With GINsim 3.0. *Frontiers in Physiology*, 9, 646. <https://doi.org/10.3389/fphys.2018.00646>
108. Chaouiya, C., Bérenguier, D., Keating, S. M., Naldi, A., Iersel, M. P. van, Rodriguez, N., Dräger, A., Büchel, F., Cokelaer, T., Kowal, B., Wicks, B., Gonçalves, E., Dorier, J., Page, M., Monteiro, P. T., Kamp, A. von, Xenarios, I., Jong, H. de, Hucka, M., ... Helikar, T. (2013). SBML qualitative models: A model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Systems Biology*, 7(1), 1–15. <https://doi.org/10.1186/1752-0509-7-135>
109. Müssel, C., Hopfensitz, M., & Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, 26(10), 1378–1380. <https://doi.org/10.1093/bioinformatics/btq124>
110. Veliz-Cuba, A., Aguilar, B., Hinkelmann, F., & Laubenbacher, R. (2014). Steady state analysis of Boolean molecular network models via model reduction and computational algebra. *BMC Bioinformatics*, 15(1), 221. <https://doi.org/10.1186/1471-2105-15-221>
111. *JUnit 5*. (2021). <https://junit.org/junit5/>, (15 May 2021, date last accessed).
112. *AssertJ - fluent assertions java library*. (2021). <https://assertj.github.io/doc/>, (15 May 2021, date last accessed).
113. *Mockito framework site*. (2021). <https://site.mockito.org/>, (15 May 2021, date last accessed).
114. Xie, Y. (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman; Hall/CRC. <https://bookdown.org/yihui/bookdown>
115. Flobak, Å., Niederdorfer, B., Nakstad, V. T., Thommesen, L., Klinkenberg, G., & Lægreid, A. (2019). A high-throughput drug combination screen of targeted small molecule inhibitors in cancer cell lines. *Scientific Data*, 6(1), 237. <https://doi.org/10.1038/s41597-019-0255-7>
116. Flobak, Å., Vazquez, M., Lægreid, A., & Valencia, A. (2017). CIminator: a web-based tool for drug synergy analysis in small- and large-scale datasets. *Bioinformatics*, 33(15), 2410–2412. <https://doi.org/10.1093/bioinformatics/btx161>
117. *KEGG PATHWAY: Pathways in cancer - Homo sapiens (human)*. (2000). https://www.genome.jp/kegg-bin/show_pathway?hsa05200, (15 May 2021, date last accessed).
118. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
119. Saito, T., & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, 10(3),

- e0118432. <https://doi.org/10.1371/journal.pone.0118432>
120. Perkins, N. J., & Schisterman, E. F. (2006). The Inconsistency of “Optimal” Cutpoints Obtained using Two Criteria based on the Receiver Operating Characteristic Curve. *American Journal of Epidemiology*, 163(7), 670–675. <https://doi.org/10.1093/aje/kwj063>
121. Chang, W., Cheng, J., Allaire, J. J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., & Borges, B. (2021). *shiny: Web Application Framework for R*. <https://cran.r-project.org/package=shiny>, R package version 1.6.0.
122. Xie, Y., Cheng, J., & Tan, X. (2021). *DT: A Wrapper of the JavaScript Library 'DataTables'*. <https://cran.r-project.org/package=DT>, R package version 0.18.
123. Sievert, C. (2020). *Interactive Web-Based Data Visualization with R, plotly, and shiny*. Chapman; Hall/CRC. <https://plotly-r.com>
124. Grau, J., Grosse, I., & Keilwagen, J. (2015). PRROC: computing and visualizing precision-recall and receiver operating characteristic curves in R. *Bioinformatics*, 31(15), 2595–2597. <https://doi.org/10.1093/bioinformatics/btv153>
125. Wickham, H. (2015). *R packages: organize, test, document, and share your code*. O'Reilly Media, Inc.
126. Roache, P. J. (1997). Quantification of uncertainty in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 29, 123–160. <https://doi.org/10.1146/annurev.fluid.29.1.123>
127. Mendoza, L., & Xenarios, I. (2006). A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theoretical Biology and Medical Modelling*, 3(1), 13. <https://doi.org/10.1186/1742-4682-3-13>
128. Fauré, A., & Thieffry, D. (2009). Logical modelling of cell cycle control in eukaryotes: A comparative study. *Molecular BioSystems*, 5(12), 1569–1581. <https://doi.org/10.1039/b907562n>
129. Niederdorfer, B., Touré, V., Vazquez, M., Thommesen, L., Kuiper, M., Lægreid, A., & Flobak, Å. (2020). Strategies to Enhance Logic Modeling-Based Cell Line-Specific Drug Synergy Prediction. *Frontiers in Physiology*, 11, 862. <https://doi.org/10.3389/fphys.2020.00862>
130. Videla, S., Saez-Rodriguez, J., Guziolowski, C., & Siegel, A. (2016). Caspo: A toolbox for automated reasoning on the response of logical signaling networks families. *Bioinformatics*, 33(6), 947–950. <https://doi.org/10.1093/bioinformatics/btw738>
131. Gjerga, E., Trairatphisan, P., Gabor, A., Koch, H., Chevalier, C., Ceccarelli, F., Dugourd, A., Mitsos, A., & Saez-Rodriguez, J. (2020). Converting networks to predictive logic models from perturbation signalling data with CellNOpt. *Bioinformatics*. <https://doi.org/10.1093/bioinformatics/btaa561>
132. Aghamiri, S. S., Singh, V., Naldi, A., Helikar, T., Soliman, S., & Niarakis, A. (2020). Automated inference of Boolean models from molecular interaction maps using CaSQ. *Bioinformatics*. <https://doi.org/10.1093/bioinformatics/btaa484>

133. Helikar, T., Kowal, B., Madrahimov, A., Shrestha, M., Pedersen, J., Limbu, K., Thapa, I., Rowley, T., Satalkar, R., Kochi, N., Konvalina, J., & Rogers, J. A. (2012). Bio-Logic Builder: A Non-Technical Tool for Building Dynamical, Qualitative Models. *PLoS ONE*, 7(10), e46417. <https://doi.org/10.1371/journal.pone.0046417>
134. Cury, J. E. R., Monteiro, P. T., & Chaouiya, C. (2019). Partial Order on the set of Boolean Regulatory Functions. *arXiv*. <http://arxiv.org/abs/1901.07623>
135. Saez-Rodriguez, J., Alexopoulos, L. G., Epperlein, J., Samaga, R., Lauffenburger, D. A., Klamt, S., & Sorger, P. K. (2009). Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Molecular Systems Biology*, 5(1), 331. <https://doi.org/10.1038/msb.2009.87>
136. Shmulevich, I., & Kauffman, S. A. (2004). Activities and sensitivities in Boolean network models. *Physical Review Letters*, 93(4), 048701. <https://doi.org/10.1103/PhysRevLett.93.048701>
137. Gherardi, M., & Rotondo, P. (2016). Measuring logic complexity can guide pattern discovery in empirical systems. *Complexity*, 21, 397–408. <https://doi.org/10.1002/cplx.21819>
138. Abou-Jaoudé, W., & Monteiro, P. T. (2019). On logical bifurcation diagrams. *Journal of Theoretical Biology*, 466, 39–63. <https://doi.org/10.1016/j.jtbi.2019.01.008>
139. Abou-Jaoudé, W., Ouattara, D. A., & Kaufman, M. (2009). From structure to dynamics: Frequency tuning in the p53-Mdm2 network. I. Logical approach. *Journal of Theoretical Biology*, 258(4), 561–577. <https://doi.org/10.1016/j.jtbi.2009.02.005>
140. McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv*. <http://arxiv.org/abs/1802.03426>
141. Zobolas, J. (2020). *Rtemp: R Templates for Reproducible Data Analyses*. GitHub. <https://github.com/bblodfon/rtemp>
142. Vázquez, M., Nogales, R., Carmona, P., Pascual, A., & Pavón, J. (2010). *Rbbt: A Framework for Fast Bioinformatics Development with Ruby* (pp. 201–208). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13214-8_26

Papers

PAPER 1

Databases and ontologies

UniBioDicts: Unified access to Biological Dictionaries

John Zobolas *, Vasundra Touré , Martin Kuiper  and Steven Vercruyse 

Department of Biology, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway

*To whom correspondence should be addressed.

Associate Editor: Peter Robinson

Received on August 1, 2020; revised on November 20, 2020; editorial decision on December 9, 2020; accepted on December 11, 2020

Abstract

Summary: We present a set of software packages that provide uniform access to diverse biological vocabulary resources that are instrumental for current biocuration efforts and tools. The Unified Biological Dictionaries (UniBioDicts or UBDs) provide a single query-interface for accessing the online API services of leading biological data providers. Given a search string, UBDs return a list of matching term, identifier and metadata units from databases (e.g. UniProt), controlled vocabularies (e.g. PSI-MI) and ontologies (e.g. GO, via BioPortal). This functionality can be connected to input fields (user-interface components) that offer autocomplete lookup for these dictionaries. UBDs create a unified gateway for accessing life science concepts, helping curators find annotation terms across resources (based on descriptive metadata and unambiguous identifiers), and helping data users search and retrieve the right query terms.

Availability and implementation: The UBDs are available through npm and the code is available in the GitHub organisation UniBioDicts (<https://github.com/UniBioDicts>) under the Afferro GPL license.

Contact: john.zobolas@ntnu.no

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Motivation

The plethora of ontology terms and biological entity identifiers (IDs) provides a vast resource for use in annotations (by curators) and in database queries (by life scientists and computers), but specifying and finding them requires extensive navigation through an intimidating number of web resources and look-up forms. A universal way to perform a comprehensive search of life science databases, ontologies and vocabularies, supported by an autocomplete function that allows users to choose from a list of candidate terms with defining metadata, will greatly streamline this process. In addition, it will help to eliminate errors that stem from typing these terms manually without autocomplete support or options for semantic input checking. Furthermore, a unified lookup utility makes terms from diverse vocabularies easy to place together into context-rich annotations. The Visual Syntax Method (VSM) for example (Vercruyse and Kuiper, 2020), a technology that allows the flexible annotation of virtually any type of contextual information, can take advantage of unified access to such a large diversity of terms, e.g. in applications like causalBuilder (Touré et al., 2020). For these reasons, we set out to create a software suite that maps many of the diverse resources to a single data access and representation form.

2 Implementation

Each UBD module is an interface to an online server that provides ontology or controlled vocabulary data. A single dictionary module may provide access to one or several apparent ‘sub-dictionaries’; e.g.

the BioPortal UBD presents each of its many combined biological-domain ontologies as a distinct sub-dictionary. When a UBD receives a request for data, it makes a custom request to the associated server’s API, and translates received data back into the format specified by the [generic dictionary interface](#).

2.1 Main methods and data-types

Each UBD module offers the following methods to access a resource’s data, along with options for filtering, sorting and paging of results:

1. `getDictInfos`: returns a list of `dictInfo` objects which each hold information about one sub-dictionary of the data resource.
2. `getEntries`: returns `entry` objects. Each `entry` represents all relevant information about a specific biological concept. It is the combination of a computer-processable ID, at least one human-friendly term (a word or word sequence), and various metadata. The combined metadata makes it possible to inform curators of what a concept represents and how its meaning differs from others. For example, the UniProt UBD returns the ‘tp53’ concept via the standard properties: `id` (a URI, Uniform Resource ID: ‘<https://www.uniprot.org/uniprot/P04637>’), `terms` (a list: ‘P53_HUMAN’, ‘Cellular tumor antigen p53’, etc., with recommended name first and synonyms next), `descr` (text description of the protein), `dictID` (URI for the resource: ‘<https://www.uniprot.org>’); and an extra set of `z` sub-properties for data

- specific to UniProt: *z.species* ('Homo Sapiens'), *z.genes* ('TP53', 'P53'), etc.
3. `getEntryMatchesForString`: returns match objects. Each match combines one term-string (which may be a synonym, for one or several entries) with a specific entry that it represents. For example, querying the UniProt dictionary for 'tumor antigen p53' returns among others the above entry object for 'tp53', augmented with the property *str* ('P53_HUMAN').

For each UBD, these 'get-' methods have been harmonized with the associated resource's available search and returned data. This is detailed in each UBD's Readme on GitHub.

2.2 Additional features

1. Several UBDs are **optimized for curator use**: a `match` object's *descr* and *str* are tweaked so that an autocomplete list can present available concepts in a way that is helpful in biocuration tasks. For example, when the Ensembl UBD queries its server for 'tp53', it receives several gene concepts with the same name and description, but different species and gene-synonyms. So to provide a more informative description, the last three are combined into an optimized *descr*.
2. Identifiers (*id*, *dictID*) are formed as **unambiguous, browsable URIs**. This supports giving users clickable access to details about a returned concept to verify if it conveys the desired semantics for their annotation (McMurtry et al., 2017).
3. UBDs `entry` objects are **extensible**. Any extra information offered by a resource's API can be added in the `entry.z` object, where it can later be used to customize or augment what an autocomplete shows to the user.

For further discussion on implementation and the expected impact of UBDs in the biocuration world, see [Supplementary File S1](#).

3 Results

3.1 Implemented UBDs

Current UBDs map and unify the following biological resources and their respective APIs:

- BioPortal (Whetzel et al., 2011), the largest repository of biomedical ontologies, using the [BioPortal REST API](#)
- PubMed MEDLINE database of biomedical literature, using the Entrez programming utilities (Sayers, 2010)
- Noctua Entity Ontology, using their Solr Web service
- UniProt (The UniProt Consortium, 2019), using their [REST API](#)
- Ensembl (Zerbino et al., 2018)
- Ensembl Genomes (Howe et al., 2020)
- RNACentral (The RNACentral Consortium, 2018)
- Complex Portal (Meldal et al., 2019)

The last four UBDs each process a different data domain from the EBI Search API (Madeira et al., 2019). In addition, we provide a [package](#) that can combine several UBDs into one virtual dictionary, enabling the querying of multiple UBDs through one access point (see [demo example](#) where a [vsm-box](#) tool's autocomplete is linked to UBDs).

3.2 Potential users

1. **Research software engineers** who use UBDs as a meta-API. They can programmatically access multiple resources in a uniform

way and avoid dealing with disparate APIs that all have different documentation, specifications and data formats.

2. **Software developers** who build a project-specific curation tool. They can create input fields that offer autocomplete lookup in any set of UBDs and present matching terms and IDs in a selection panel. This is easily achieved by linking any dictionary to our reusable [autocomplete web-component](#). UBDs can also be linked to a vsm-box (Vercruyse et al., 2020) to build curation applications, like causalBuilder.
3. **Biocurators** who use the above curation tools to find the terms they need. Autocomplete-based annotation allows biocurators to curate papers more quickly, conveniently and precisely, without having to copy text and IDs from elsewhere (Ward et al., 2012).

Acknowledgements

The authors thank all the developers of the various data sources and web services whom they consulted during the design and implementation of this work. Special thanks go to Michael Dorf and Jennifer Leigh Vendetti from BioPortal, for answering a series of long emails. They also thank EMBL-EBI software engineers Youngmi Park (EBI Search), Blake Sweeney (RNACentral), Leonardo Gonzales (UniProt), Noemi Del Toro Ayllón (Complex Portal) and Kieron Taylor (Ensembl) for face-to-face discussions and support; and Berkeley scientist Laurent-Philippe Albou (Noctua Entity Ontology) for email feedback.

Funding

This work was supported by ERACoSysMed Call 1 project COLOSYS (V.T., J.Z., M.K.), the COST action Gene Regulation Ensemble Effort for the Knowledge Commons [CA15205] (V.T., J.Z., M.K., S.V.), the Norwegian University of Science and Technology's Strategic Research Area 'NTNU Health' (VT), the Research Council of Norway [247727/O70] (S.V.) and S.V. [2020].

Conflict of Interest: none declared.

References

- Howe,K.L. et al. (2020) Ensembl Genomes 2020-enabling non-vertebrate genomic research. *Nucleic Acids Res.*, 48, D689–D695.
- Madeira,F. et al. (2019) The EMBL-EBI search and sequence analysis tools APIs in 2019. *Nucleic Acids Res.*, 47, W636–W641.
- McMurtry,J.A. et al. (2017) Identifiers for the 21st century: how to design, provision, and reuse persistent identifiers to maximize utility and impact of life science data. *PLoS Biol.*, 15, e2001414.
- Meldal,B.H. et al. (2019) Complex Portal 2018: extended content and enhanced visualization tools for macromolecular complexes. *Nucleic Acids Res.*, 47, D550–D558.
- Sayers,E. (2010) *Entrez Programming Utilities Help*. <https://www.ncbi.nlm.nih.gov/books/NBK25501/> (12 December 2020, date last accessed).
- The RNACentral Consortium. (2018) RNACentral: a hub of information for non-coding RNA sequences. *Nucleic Acids Res.*, 47, D1250–D1251.
- The UniProt Consortium. (2019) UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.*, 47, D506–D515.
- Touré,V. et al. (2020) CausalBuilder: bringing the MI2CAST causal interaction annotation standard to the curator. *Preprints*. doi:10.20944/preprints202007.0622.v1.
- Vercruyse,S. and Kuiper,M. (2020) Intuitive representation of computable knowledge. *Preprints*. doi:10.20944/preprints202007.0486.v2.
- Vercruyse,S. et al. (2020) VSM-box: general-purpose interface for biocuration and knowledge representation. *Preprints*. doi: 10.20944/preprints202007.0557.v1.
- Ward,D. et al. (2012) Autocomplete as a research tool: a study on providing search suggestions. *Inf. Technol. Libraries*, 31, 6–19.
- Whetzel,P.L. et al. (2011) BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Res.*, 39, W541–5.
- Zerbino,D.R. et al. (2018) Ensembl 2018. *Nucleic Acids Res.*, 46, D754–D761.

Author's comment: In this supplementary file, we discuss what UBDs can offer to the biocuration and systems biology world, and the problems that we faced and had to overcome towards this goal.

Discussion

Although many of the leading resources provide individual support for finding appropriate identifiers, terms and definitions for biological entities and concepts, an overarching function that spans all resources is not yet available. Such a utility, providing real-time access to terminology from diverse biological subdomains through a unified interface, enables the development of tools that build upon the collective information residing in these disparate domains. A unified access to the wealth of descriptive information forms an essential enabling part of computational, semantic systems biology. Continuing in this spirit, we have recently started building another [UBD](#) that connects with PubDictionaries (Kim et al. 2019), and we invite future collaborators to join our [UniBioDicts](#) GitHub organisation and help build a growing collection of client APIs serving biological dictionaries. The currently developed packages cover a diverse range of web-services, API-technologies and associated data-types, providing concrete examples that facilitate the development of additional UBDs, or for that matter, any other domain dictionaries that may need to access online databases or ontologies for curation.

In the process of building the UBDs, we had to consult with at least one developer from each data or API resource, in order to clarify, refine, and simplify both their and our documentation and specification details, which subsequently led to a better design of the software. For example, individual APIs return error objects in different ways, which prompted us to harmonize our error handling specification across all UBDs. In order to deliver robust software that will benefit its users and optimize software development efforts in the future, face-to-face discussions coupled with extensive Q&A email correspondence proved to be essential (Prlić and Procter 2012). Finally, we wish to emphasize the importance that proper documentation has in a healthy software development practice (Karimzadeh and Hoffman 2018), and its vital role in achieving our aforementioned goal.

References

- Karimzadeh, Mehran, and Michael M. Hoffman. 2018. “Top Considerations for Creating Bioinformatics Software Documentation.” *Briefings in Bioinformatics* 19 (4): 693–99.
- Kim, Jin-Dong, Yue Wang, Toyofumi Fujiwara, Shujiro Okuda, Tiffany J. Callahan, and K. Bretonnel Cohen. 2019. “Open Agile Text Mining for Bioinformatics: The PubAnnotation Ecosystem.” *Bioinformatics* 35 (21): 4372–80.
- Prlić, Andreas, and James B. Procter. 2012. “Ten Simple Rules for the Open Development of Scientific Software.” *PLoS Computational Biology* 8 (12): e1002802.

PAPER 2

Linking PubDictionaries with UniBioDicts to support Community Curation

John Zobolas^{1, *}, Jin-Dong Kim^{2, *}, Martin Kuiper¹, and Steven Vercruyse³

1 Department of Biology, Norwegian University of Science and Technology (NTNU), Trondheim, Norway **2** Database Center for Life Science (DBCLS), Tokyo, Japan **3** Independent Scientist, Trondheim, Norway * Shared first authorship

Abstract

One of the many challenges that biocurators face, is the continuous evolution of ontologies and controlled vocabularies and their lack of coverage of biological concepts. To help biocurators annotate new information that cannot yet be covered with terms from authoritative resources, we produced an update of **PubDictionaries**: a resource of publicly editable, simple-structured dictionaries, accessible through a dedicated REST API. PubDictionaries was equipped with both an enhanced API and a new software client that connects it to the **Unified Biological Dictionaries** (UBDs) uniform data exchange format. This client enables efficient search and retrieval of *ad hoc* created terms, and easy integration with tools that further support the curator's specific annotation tasks. A demo that combines the Visual Syntax Method (VSM) interface for general-purpose knowledge formalization, with this new PubDictionaries-powered UBD client, shows it is now easy to incorporate the user-created PubDictionaries terminologies into biocuration tools.

Introduction

The curation of biological information is met with several challenges today. The constant refactoring of ontologies, nomenclature and identifiers, as well as the discovery of new types of information and new uses thereof, makes the life of knowledge curators difficult, especially in the highly diverse domain of biology. For example, expert curators who rely on specialized software tools in the annotation process might come across a new concept or feature that does not exist within an ontology or vocabulary that their annotation tool connects to. Similar difficulties are faced by non-expert curators. Some biologists want to create a project-specific knowledge resource in a biological niche that is only minimally or not yet covered by existing controlled vocabularies. Under time pressure of project milestones, these biologists may not immediately have the resources to organize a multilateral effort to standardize the terminology in their field. Instead they typically resort to creating 'private' vocabularies within their projects, that may later serve as a first step toward larger-scale coordination (Hartmann et al., 2019).

In cases where there is an abundance of resources for controlled vocabularies, ontologies and identifiers, it may still be challenging to coordinate access to these many necessary resources for dedicated annotation endeavours. Alternatively, in cases where no proper controlled vocabulary would exist, the results from all the work that goes into creating new vocabularies will remain largely isolated from general use, if no term sharing mechanism is available. Existing tools would then also be unable to access these newly created terms and for example serve them to curators in an autocomplete panel, to make their task easier and less error-prone.

These problems are addressed by two complementary initiatives. On the one hand, **Unified Biological Dictionaries (UniBioDicts or UBDs)** (Zobolas, Touré, Kuiper, & Vercruyse, 2020) is a set of software packages that offers a unified, single access point to biological terminology resources, and is available to be plugged into any curation platform currently in operation (for example the general-purpose curation interface ‘vsm-box’ (Vercruyse et al., 2020), based on VSM (Vercruyse & Kuiper, 2020), is out-of-the-box compatible with UBDs). On the other hand, **PubDictionaries** (Kim et al., 2019) is a public repository of dictionaries where users can create and immediately share their own dictionaries based on a simple data format consisting of a term and an identifier.

During the ELIXIR BioHackathon 2020, we updated and improved the PubDictionaries API as well as developed a new UBD client that directly communicates with that API. These implementation efforts constitute a significant step towards the unification of some of the most important data resources across all biological domains. The addition of PubDictionaries to the list of UniBioDicts-interoperable resources now provides a uniform way to search and autocomplete terms from all these community-created dictionaries as well. Such functionality enables the easier integration of PubDictionaries in any curation tool that may have a need for their terms and for such a term suggestion feature.

Results

In the following two subsections, we briefly summarize the results of the hackathon efforts, grouped into two categories:

- Updating the PubDictionaries REST API
- Creating a new UBD client library to access the above API

PubDictionaries REST API

PubDictionaries is a public repository of dictionaries, where each dictionary is a collection of **labels** (human-friendly **terms**) and **identifiers** (unambiguous **IDs**, used by computers). Each label + ID pair is called an **entry**. Multiple entries can have the same ID (for synonymous labels) and the same label can occur in multiple entries (for ambiguous ones). Users can create their own dictionaries and add entries to them via the web-interface. The dictionaries can be used to annotate any piece of text via the PubAnnotation ecosystem (Kim et al., 2019) or to simply lookup terms, and both these services are supported by a RESTful API (Kim, 2020). All the API responses are structured as JSON objects. Prior to the BioHackathon event, the REST service provided the following main endpoints:

1. **find_ids**: given some specific terms and dictionary names, this endpoint returns the corresponding IDs that approximately match the terms in these dictionaries. Example: https://pubdictionaries.org/find_ids.json?labels=TP53&dictionaries=human-UniProt
2. **prefix_completion/substring_completion**: given a term (or partial term string), these endpoints search for prefix (respectively substring) matches in a specified dictionary and return the corresponding entries. Note that only a first page of results was returned with at most 15 entries, prior to the hackathon efforts. Example: https://pubdictionaries.org/dictionaries/human-UniProt/prefix_completion?term=p53
3. **text_annotation**: given a piece of text and dictionary names, this endpoint returns the result of annotation to the text using the dictionaries. The annotation is performed based on computation of string similarity between dictionary entries and expressions in the text. Example: [https://pubdictionaries.org/text_annotation.json?dictionary=human-UniProt&text=The%20tumor%20suppressor%20p53%20\(TP53\)%20is%20the%20most%20frequently%20mutated%20human%20gene](https://pubdictionaries.org/text_annotation.json?dictionary=human-UniProt&text=The%20tumor%20suppressor%20p53%20(TP53)%20is%20the%20most%20frequently%20mutated%20human%20gene)

The following REST Service endpoints were added during the BioHackathon:

1. **dictionaries** endpoint: returns information about a specific dictionary, such as its id, name, a text description, the number of entries it has, etc. Example: <https://pubdictionaries.org/dictionaries/human-UniProt.json>, where human-UniProt can be any existing dictionary name.
2. **entries** endpoint: returns all entries of a specific dictionary, paginated and sorted by label. Example: https://pubdictionaries.org/dictionaries/human-UniProt/entries.json?page=1&per_page=15 Note that the users (software developers) can explicitly configure how the result should be paginated, i.e. how many entries of a dictionary should be included in one 'result-page', and what page they want to get results back from.
3. **find_terms** endpoint: this is the complement of the **find_ids** endpoint in the sense that it returns a list of terms and dictionary names that match the given IDs. The result is first sorted by ID and then by dictionary name. If no dictionary name is given to this endpoint, then it searches for the given IDs in all dictionaries. Example: https://pubdictionaries.org/find_terms.json?dictionaries=&ids=https://www.uniprot.org/uniprot/P04637
4. **mixed_completion** endpoint: a combined and updated version of the **prefix_completion** and **substring_completion** endpoints. For a given term (or partial term string) and a specified dictionary it returns a list of entries, putting the prefix completions in the top half and the substring completions in the bottom half, while pruning any possible common entries. In addition, this endpoint supports pagination which is a direct result of extending the prefix and substring endpoints to support this feature as well. Example: https://pubdictionaries.org/dictionaries/human-UniProt/mixed_completion?term=p53&page=2&per_page=3

Additional work on the PubDictionaries server-side included the support of create (via the HTTP POST method) and delete operations of a specific dictionary, given certified user credentials.

Lastly, the error handling was harmonized across all REST URL endpoints. In particular, when a user searches for a non-existent dictionary name, the PubDictionaries server returns a proper HTTP response status code, 400 (Bad Request), together with a JSON-formatted description as follows: { "message": "Unknown dictionary: <name>." }. For example, all the following URL links return such a response object:

- <https://pubdictionaries.org/dictionaries/non-existent-dictionary-name.json>
- https://pubdictionaries.org/find_terms.json?dictionaries=non-existent-dictionary-name&ids=id1,id2
- <https://pubdictionaries.org/dictionaries/non-existent-dictionary-name/entries.json>

UBD Client for PubDictionaries

UBDs are a set of software packages that provide a unified query-interface for accessing the online API services of key biological vocabulary-data providers (Zobolas et al., 2020). The main feature of UBDs is their string-search functionality, which returns for a given label (or partial label) a list of matching term, identifier and metadata units from databases (e.g. UniProt (The UniProt Consortium, 2019)), controlled vocabularies (e.g. PSI-MI), and ontologies (e.g. Gene Ontology, via BioPortal (Whetzel et al., 2011)). This feature makes UBDs ideal for enabling autocomplete support in user-interface components that serve terms to curators from disparate resources, thus allowing the more efficient annotation of information.

Our work in the ELIXIR BioHackathon 2020 included the creation of a new UBD client ([vsm-pubdictionaries](#)) that utilizes the updated PubDictionaries API in order to solve a long-standing problem in the biocurator community: how can *ad hoc*, project-specific terms and new information be effectively annotated with, and served via a curation platform, without the need to first negotiate the storage, update and reconciliation of that information with a third party, e.g. a database or ontology provider? Our client software addresses this problem

by presenting a mediator solution that can easily be plugged into current curation applications and serve *ad hoc* terms from PubDictionaries' public curator-created dictionaries.

Regarding the software client code, we wrote [extensive documentation](#) to delineate the mapping between the terms and IDs from PubDictionaries and the unified UBD format and how this is achieved via the updated PubDictionaries REST API endpoints, all in accordance with the UBDs' shared [dictionary interface specification](#). We also enabled continuous integration support via [GitHub Actions](#) and wrote extensive tests ([code coverage](#) is at 95%), so as to deliver more reliable, fault-tolerant and easy-to-extend software. Moreover, the documentation includes two examples; one showcasing the search term functionality via `Node.js` and one indicating how to use the client library in a web-based environment, with an HTML file. Finally, the [demo example](#) (see Figure 1) that was presented during the last report session of the BioHackathon, demonstrates a simple use-case where a few public dictionaries were created and their terms served in a vsm-box curation interface (Vercruyse et al., 2020). Thus we show how straightforward the annotation of new information can be by means of the autocomplete functionality of the provided curation tool, and how this new knowledge can be connected with semantically aware annotations.

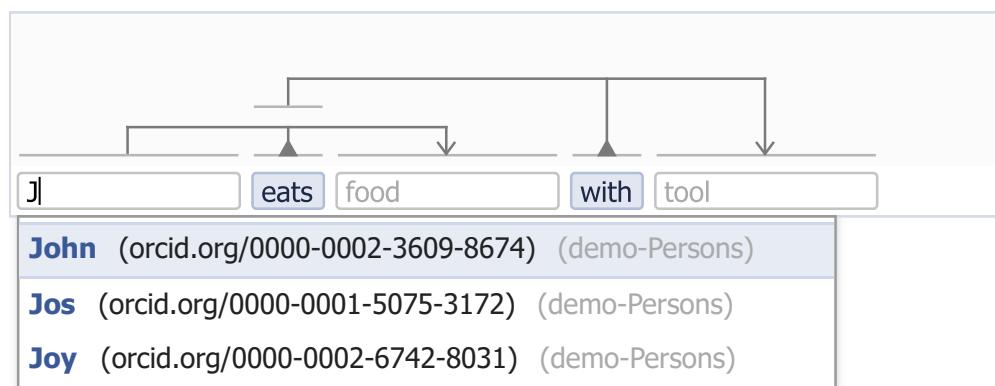


Figure 1: Demo example that uses a 'vsm-box' curation interface component, pre-filled with a VSM-template. An autocomplete panel appears while the user enters terms linked to identifiers. These term+ID pairs come from demo-dictionaries that we created at PubDictionaries, and are fetched through the new REST API-client described in this paper. Placeholders like 'food' and 'tool' indicate the kinds of dictionaries that specific fields of the template are connected with. On top, VSM-connectors formalize the structure and meaning of this knowledge unit.

Discussion

The advantages of a software package that connects with and queries any dictionary created in the PubDictionaries web-interface are multiple. This novel software enables annotation tools to use a common language and interface to link to information that is not yet available in standard databases. Note that the process of integrating new terms into standard resources can be time-consuming, so supporting communities of curators to create and utilize new terms that are at least publicly shared, in PubDictionaries, is a helpful first measure to tackle the problem of missing terms during ongoing curation work. Our software is a step towards achieving that goal, since it positions the community-manageable PubDictionaries into the mainstream of controlled vocabularies (CVs) and ontology resources. It fills the niche of new and *ad hoc* CVs that in turn may prompt new dedicated efforts to mature these CVs for consensus and expert maintenance.

Propelled by an ELIXIR BioHackathon event, our work underpins the goals of several of ELIXIR's activities, or so-called 'Platforms'. In the ELIXIR Data Platform, the drive to use, re-use and value life science data takes precedence. Our efforts exemplify how to achieve this by providing a scalable solution for curation platforms, especially ones that include support for annotation efforts that involve new information types. Furthermore, our main objective matches the goals of the ELIXIR Interoperability Platform: we offer a way to publicly-access and integrate new curated knowledge in a unified form, which enables new knowledge to be used by humans and machines alike, and to build knowledge systems that will aid future endeavors in understanding biological processes.

Future Work

Our future work includes updates on the PubDictionaries API to support the addition, update and deletion of dictionary entries, which is a functionality currently only available in the web-interface of PubDictionaries. Consequently, a further update on the UBD client will provide the necessary backbone to help build user interfaces, where curators would not even need to log in to the PubDictionaries website to create new dictionaries, and add, update or delete entries, but rather would be able to do that from within their own in-house curation tool. This functionality is currently not offered by any other biological data provider. Lastly, we expect that these updates, coupled with the search-string functionality provided by the PubDictionaries UBD client, will contribute in efforts to significantly increase the autonomy of biocurators and their potential for creating shareable annotations.

Links to software and documentation

- PubDictionaries API documentation: <https://docs.pubdictionaries.org>
- PubDictionaries source code: <https://github.com/pubannotation/pubdictionaries>
- UBD Client for PubDictionaries: <https://github.com/UniBioDicts/vsm-pubdictionaries>
- Demo example with vsm-box curation interface: https://github.com/UniBioDicts/vsm-pubdictionaries/blob/master/test/test_vsm_box_pubdictionaries.html

Acknowledgements

This work was carried out during the virtual Europe BioHackathon event that was organized by ELIXIR in November 2020. We would like to thank the organizers for the excellent management of such a large-scale, virtual event with 200+ participants and for creating the opportunity to meet, discuss, collaborate and share ideas and technologies with many new people.

References

- Hartmann, N. B., Hüffer, T., Thompson, R. C., Hassellöv, M., Verschoor, A., Daugaard, A. E., Rist, S., et al. (2019). Are We Speaking the Same Language? Recommendations for a Definition and Categorization Framework for Plastic Debris. *Environmental Science and Technology*, 53(3), 1039–1047. doi:[10.1021/acs.est.8b05297](https://doi.org/10.1021/acs.est.8b05297)
- Kim, J.-D. (2020). PubDictionaries REST API documentation. Retrieved from <https://docs.pubdictionaries.org/>
- Kim, J.-D., Wang, Y., Fujiwara, T., Okuda, S., Callahan, T. J., & Cohen, K. B. (2019). Open Agile text mining for bioinformatics: the PubAnnotation ecosystem. *Bioinformatics*, 35(21), 4372–4380. doi:[10.1093/bioinformatics/btz227](https://doi.org/10.1093/bioinformatics/btz227)
- The UniProt Consortium. (2019). UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1), D506–D515. doi:[10.1093/nar/gky1049](https://doi.org/10.1093/nar/gky1049)



Vercruyse, S., & Kuiper, M. (2020). Intuitive representation of computable knowledge. *Preprints*. doi:[10.20944/preprints202007.0486.v2](https://doi.org/10.20944/preprints202007.0486.v2)

Vercruyse, S., Zobolas, J., Touré, V., Andersen, M. K., & Kuiper, M. (2020). VSM-box: general-purpose interface for biocuration and knowledge representation. *Preprints*. doi:[10.20944/preprints202007.0557.v1](https://doi.org/10.20944/preprints202007.0557.v1)

Whetzel, P. L., Noy, N. F., Shah, N. H., Alexander, P. R., Nyulas, C., Tudorache, T., & Musen, M. A. (2011). BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic acids research*, 39(Web Server issue), W541–5. doi:[10.1093/nar/gkr469](https://doi.org/10.1093/nar/gkr469)

Zobolas, J., Touré, V., Kuiper, M., & Vercruyse, S. (2020). UniBioDicts: Unified access to Biological Dictionaries. *Bioinformatics*. doi:[10.1093/bioinformatics/btaa1065](https://doi.org/10.1093/bioinformatics/btaa1065)

PAPER 3

Authors

Asmund Flobak^{1,2}, John Zobolas¹, Miguel Vazquez, Tonje Steigedal, Liv Thommesen, Asle Grislingås, Barbara Niederdorfer, Evelina Folkesson, Martin Kuiper

¹ These authors contributed equally

² Corresponding author

Author contributions

ÅF: Devised the project and developed the main conceptual ideas. Designed and developed prototype for simulation pipeline. Developed and implemented drug response combination simulation software. Analysed drug response prediction data and model performance. Designed and executed cell line experiments. Manually curated drug combination response data. Designed and executed xenograft experiment. Supervised drug combination matrix experiments. Supervised tissue microarray experiments.

JZ: Developed and implemented production-ready drug response combination simulation software. Executed various simulations, analysed drug response prediction data and model performance, produced resulting figures, produced supplementary information, produced github repositories, made all analyses reproducible.

MV: Analysed drug combination data. Developed and implemented drug response combination simulation software.

TS: Supervised xenograft experiments.

LT: Supervised xenograft experiments.

AG: Prepared and analysed tissue microarrays of xenograft tumours.

BN: Manually curated drug combination response data. Designed and executed drug combination matrix experiments.

EF: Manually curated drug combination response data. Designed and executed drug combination matrix experiments.

MK: Analysed drug response prediction data and model performance. Supervised the project. All authors contributed to the manuscript.

Affiliations

ÅF1: Department of Clinical and Molecular Medicine, Norwegian University of Science and Technology, Trondheim, Norway.

ÅF2: The Cancer Clinic, St. Olav's Hospital, Trondheim, Norway.

JZ: Department of Biology, Norwegian University of Science and Technology, Trondheim, Norway.

MV:

TS:

LT:

AG:

BN:

EF:

MK: Department of Biology, Norwegian University of Science and Technology, Trondheim, Norway

Title

Fine tuning a logical model of cancer cells to predict drug synergies:
combining manual curation and automated parameterization

Abstract

Treatment with combinations of drugs carries great promise for personalized therapy for a variety of diseases. We have previously shown that synergistic combinations of cancer signaling inhibitors can be identified based on a logical framework by manual model definition. We now demonstrate how automated adjustments of model topology and logic equations both can greatly reduce the workload traditionally associated with logical model optimization. Our methodology allows the exploration of larger model ensembles that all obey a set of observations, while being less restrained for parts of the model where parameterization is not guided by biological data. We benchmark the synergy prediction performance of our logical models in a dataset of 153 targeted drug combinations. We show that well-performing manual models faithfully represent measured biomarker data and that their performance can be outmatched by automated parameterization using a genetic algorithm. Whereas the predictive performance of a curated model is strongly affected by simulated curation errors, data-guided deletion of a small subset of regulatory model edges can significantly improve prediction quality. With correct topology we find evidence of some tolerance to simulated errors in the biomarker calibration data, yet performance decreases with reduced data quality. Moreover, we show that predictive logical models are valuable for proposing mechanisms underpinning observed synergies. With our framework we predict the synergy of joint inhibition of PI3K and TAK1, and further substantiate this prediction with observation in cancer cell cultures and in xenograft experiments.

Introduction

Combining specific and targeted drugs in one therapy to fight disease increases chances of treatment success¹. Drug combinations that together act in synergy are especially attractive because they allow for pushing treatment effects beyond those obtainable by each drug alone², with drug dosages that can be well below levels where individual drugs begin to cause adverse effects. In addition, synergistic drug combinations may have reduced side-effects by improved selectivity in a specific biological context, for instance by allowing targeting of only certain cell types in an organism³. Lastly, searching for new combination therapies has the additional benefit that already approved drugs can act beneficially in novel combinations, and thus even allow bypassing initial drug development phases.

While the development of rational drug combination treatment has become a major priority due to hopes of increased treatment potency, a grand challenge remains in dealing with their identification in the vast potential drug combination space. Currently, more than two hundred

¹ Hanahan, D. & Weinberg, R. a. Hallmarks of cancer: the next generation. *Cell* 144, 646–74 (2011).

² Al-Lazikani, B., Banerji, U. & Workman, P. Combinatorial drug therapy for cancer in the post-genomic era. *Nat. Biotechnol.* 30, 679–92 (2012)

³ Lehár, J. et al. Synergistic drug combinations tend to improve therapeutically relevant selectivity. *Nat. Biotechnol.* 27, 659–66 (2009)

drugs have been approved by the FDA to treat cancer (<https://www.cancer.gov/about-cancer/treatment/drugs>). The testing of drugs in all combinations with other drugs in a panel needs assays in numbers that increase exponentially with increasing drug numbers. Even a modestly sized drug panel of 150 drugs corresponds to over 10.000 pairwise drug combinations. Testing high numbers of drug combinations in high throughput screens on cell lines or other patient-specific model systems is costly and at some point prohibitively expensive and cumbersome. Therefore, help is needed from *in silico* drug effect simulations to produce high quality predictions that can guide drug combination screens or therapy choices for testing in cell lines or patients. *In silico* simulations may help identify those combinations that are unlikely to produce synergies, which can be of significant help to reducing the large experimental search space that otherwise would need to be covered in exhaustive screens. As many drug synergies can be seen as emergent properties arising from molecular causality networks, analytical frameworks from computational systems biology seem to be well suited to the task.

Several mathematical frameworks have already been tested to mechanistically model drug combination effects, including continuous, discrete, and hybrid modeling approaches. Published approaches generally depend on molecular causalities downloaded from prior knowledge databases, extracted from large-scale data, or obtained by a combination of the two. Based on a dataset capturing proteomic responses of 14 targeted drugs, Miller et al. used ordinary differential equations to study mechanisms of synergy between inhibitors of CDK4 and IGF1R, revealing that the mechanisms rely on the activity of AKT⁴. Nelander et al. explored ODE models derived from observations on phospho-proteins and cell cycle markers following 21 pairwise applications of targeted drugs, with the aim to use best-performing pairs for design of new combination therapies⁵. In a semi-qualitative modeling approach, Klinger et al. used a perturbation dataset for MAPK, PI3K and NF-κB signaling to inform a model showing that combined inactivation of MEK and EGFR could inactivate endpoints of RAS, ERK and AKT signaling⁶. Jin et al. explored enhanced Petri nets to describe molecular processes for the synergy of an EGFR inhibitor (gefitinib) with chemotherapy (docetaxel) and identified KRT8 as a candidate gene to explain the synergy⁷. However, all of these approaches rely on extensive and costly combinatorial drug perturbation data, be it transcriptomic, proteomic, viability etc., for describing mechanisms of synergy, and therefore they require vast investments in data production and do not provide a feasible solution for the testing of the large drug combination space.

In order to reduce dependence on *a priori* perturbation experiments, attempts have been made to predict drug synergies from data obtained in a marginal experimental search space, rather

⁴ Miller, M. L. et al. Drug Synergy Screen and Network Modeling in Dedifferentiated Liposarcoma Identifies CDK4 and IGF1R as Synergistic Drug Targets. *Sci. Signal.* 6, ra85 (2013)

⁵ Nelander, S. et al. Models from experiments: combinatorial drug perturbations of cancer cells. *Mol. Syst. Biol.* 4, 216 (2008)

⁶ Klinger, B. et al. Network quantification of EGFR signaling unveils potential for targeted combination therapy. *Mol. Syst. Biol.* 9, 673 (2013)

⁷ Jin, G., Zhao, H., Zhou, X. & Wong, S. T. C. An enhanced Petri-net model to predict synergistic effects of pairwise drug combinations from gene microarray data. *Bioinformatics* 27, i310–6 (2011)

than the full combinatorial space. Fröhlich et al. used ODE models informed by transcriptomic and viability data to predict drug combination responses, finding that highly accurate predictions could be produced for those drugs for which they had viability response data⁸. In the DREAM7 - NCI-DREAM, Drug Sensitivity and Drug Synergy Challenges^{9,10,11} (NCI-DREAM), pairwise drug responses were predicted from response data obtained for each drug alone. The best performing teams in the NCI-DREAM challenge obtained a probabilistic concordance (PC) index of 0.61, on a scale ranging from 0.9 (perfect prediction) to 0.1 (perfect opposite prediction). Although this is better than random (PC index of 0.5), it clearly illustrates that obtaining accurate synergy predictions is far from trivial, due to a variety of reasons that will be discussed in this paper. In the more recent AstraZeneca-Sanger Drug Combination Prediction DREAM Challenge¹² (AZS-DREAM), one of the aims was to develop and demonstrate drug combination response predictability independent of extensive perturbation data. The development of such powerful prediction approaches has the advantage of being relevant not only to preclinical drug screens, but also to bed-side applications. Drug perturbation data clearly will not be trivial to obtain for individual patients, unless patient-derived experimental assays that mimic patient responses can be developed (e.g. xenografts, explants etc.). Sobering results from the AZS-DREAM challenge showed that most teams had balanced accuracies of 0.5-0.6, with the best performing team obtaining a balanced accuracy of only 0.69.

With the availability of training data, machine learning algorithms have also been explored to predict drug synergies^{13,14,15}. However, major limitations of such approaches are the lack of mechanistic insight¹⁶, and dependence on high quantities of training data. Despite some increase in their availability, large scale datasets on drug responses for machine learning to predict combination effects are still largely missing, in part due to great experimental complexity

⁸ Fröhlich, F., Kessler, T., Weindl, D., Shadrin, A., Schmiester, L., Hache, H., ... Hasenauer, J. (2018). Efficient Parameter Estimation Enables the Prediction of Drug Response Using a Mechanistic Pan-Cancer Pathway Model. *Cell Systems*, 7, 1–13. <http://doi.org/10.1016/j.cels.2018.10.013>

⁹ Bansal, M. et al. A community computational challenge to predict the activity of pairs of compounds. *Nat. Biotechnol.* (2014). doi:10.1038/nbt.3052

¹⁰ Sun, Y. et al. Combining genomic and network characteristics for extended capability in predicting synergistic drugs for cancer. *Nat. Commun.* 6, 8481 (2015).

¹¹ Goswami, C. P., Cheng, L., Alexander, P. S., Singal, a & Li, L. A New Drug Combinatory Effect Prediction Algorithm on the Cancer Cell Based on Gene Expression and Dose-Response Curve. *CPT pharmacometrics Syst. Pharmacol.* 4, e9 (2015).

¹² Menden, M. P., Wang, D., Mason, M. J., Szalai, B., Bulusu, K. C., Guan, Y., ... Saez-Rodriguez, J. (2019). Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. *Nature Communications*, 10(1), 2674. <http://doi.org/10.1038/s41467-019-09799-2>

¹³ Preuer, K., Lewis, R. P. I., Hochreiter, S., Bender, A., Bulusu, K. C., & Klambauer, G. (2017). DeepSynergy: Predicting anti-cancer drug synergy with Deep Learning. *Bioinformatics* (Oxford, England), (December). doi:10.1093/bioinformatics/btx806

¹⁴ Gayvert, K. M., Aly, O., Platt, J., Bosenberg, M. W., Stern, D. F., & Elemento, O. (2017). A Computational Approach for Identifying Synergistic Drug Combinations. *PLoS Computational Biology*, 13(1), 1–11. doi:10.1371/journal.pcbi.1005308

¹⁵ Tang, J., Gautam, P., Gupta, A., He, L., Timonen, S., Akimov, Y., ... Aittokallio, T. (2019). Network pharmacology modeling identifies synergistic Aurora B and ZAK interaction in triple-negative breast cancer. *Npj Systems Biology and Applications*, 5(1). <http://doi.org/10.1038/s41540-019-0098-z>

¹⁶ Yu, M. K., Ma, J., Fisher, J., Kreisberg, J. F., Raphael, B. J., & Ideker, T. (2018). Visible Machine Learning for Biomedicine. *Cell*, 173(7), 1562–1565. <http://doi.org/10.1016/j.cell.2018.05.056>

and high economic cost. Whereas efficient *in silico* therapy based on patient-specific models ultimately should be integrated into the clinical decision process, here we further investigate the performance of logical modeling of drug response of cancer cell lines. In order to reduce the dependency on large training datasets, we explore the use of cell lines measurements of biomolecules obtained at a single time point at steady state proliferation. To reduce data dependency and to improve mechanistic insights, these measurements are combined with prior knowledge to construct logical model ensembles to simulate drug combination effects. Logical model building is known to require meticulous involvement of curators and bioinformaticians, with substantial commitment to manual tinkering of models before the behavior of one model matches that of its experimental counterpart. We have previously published logical models of cancer cell lines, named CASCADE 1.0, CASCADE 2.0 and CASCADE 3.0, which demonstrate the potential of logical modeling for the prediction of drug synergies^{17, 18, 19}. However, the curation effort required to assemble a cancer signaling network and the dedicated interactive efforts needed to optimally modify logical rule definitions becomes a clear obstacle when constructing larger models.

If patient specific logical models are to be used routinely, such logical models should be trivial to construct for any cell line or patient-derived cell culture, and for any repertoire of targeted drugs. We therefore set out to automate processes required to calibrate a logical model from a set of molecular causative statements, i.e. a prior knowledge network. A software pipeline developed to that end would have to 1) assemble a network topology from structured data obtained from prior knowledge databases, 2) interpret baseline cancer cell line biomarker data into a signaling entity activity score, 3) calibrate generic logical models, created from prior knowledge data, by modifying logic equations to match the observed activity scores, and 4) predict phenotypic consequences of combinatorial interventions to the simulated model behavior. Our software solution for realizing points 3 and 4 is available at <https://github.com/druglogics>. We use a genetic algorithm to automatically parameterize a set of logic equations representing cancer growth-promoting signaling in the AGS gastric adenocarcinoma cell line. We demonstrate our approach by reproducing results from a previous manual effort and next test its utility with a larger model that was benchmarked against a dataset from a drug effect screen of 153 drug combinations. Experiments that simulate different levels of curation quality and biomarker data quality indicate the need for a reliable PKN, while still allowing for model improvement by network link pruning and parameter optimization.

¹⁷ Flobak, Å., Baudot, A., Remy, E., Thommesen, L., Thieffry, D., Kuiper, M., & Lægreid, A. (2015). Discovery of Drug Synergies in Gastric Cancer Cells Predicted by Logical Modeling. *PLoS Computational Biology*, 11(8), e1004426. <https://doi.org/10.1371/journal.pcbi.1004426>

¹⁸ Niederdorfer, B., Touré, V., Vazquez, M., Thommesen, L., Kuiper, M., Lægreid, A., & Flobak, Å. (2020). Strategies to Enhance logical modeling-Based Cell Line-Specific Drug Synergy Prediction. *Frontiers in Physiology*, 11(July), 862. <http://doi.org/10.3389/fphys.2020.00862>

¹⁹ Tsirvouli, E., Touré, V., Niederdorfer, B., Vázquez, M., Flobak, Å., & Kuiper, M. (2020). A Middle-Out Modeling Strategy to Extend a Colon Cancer Logical Model Improves Drug Synergy Predictions in Epithelial-Derived Cancer Cell Lines. *Frontiers in Molecular Biosciences*, 7, 502573. <http://doi.org/10.3389/fmbo.2020.502573>

Methods

Logical modeling

Logical models rely on the formalism initially proposed by Stuart Kauffman²⁰ and René Thomas²¹. Their approach first defines a regulatory graph consisting of nodes representing signaling entities (model components), and signed and directed edges representing regulatory interactions that connect signaling entities. The activities of all model components are then associated with the Boolean values ‘True’ and ‘False’, represented by 1 and 0, corresponding to activity and inactivity, respectively. This dichotomy of activity levels can be interpreted as activity being above or below a “threshold”: a component is “active” when its activity level is sufficiently high to influence a target component’s activity levels. In model simulations, specific model components can be defined as ‘output nodes’, whose activity values serve as a proxy for a phenotype of interest. This allows us to compute an overall ‘growth’ value in our simulations, by integrating all activity values of model output nodes, and scaling this sum to [0..1]. For example, if the anti-survival model output nodes CASP8, CASP9 and FOXO3 are inactive and the pro-survival model output nodes MYC, CCND1 and TCF7 are active, the global output ‘growth’ is 1.0. Inhibitory effects of a drug are simulated by fixing the drug target to an activity level of 0, i.e. simulating a block in signaling activity of the drug target node.

Model attractors were determined using the software bioLQM, which provides a fast algorithm for finding stable state(s) and complex attractors²². For some of the simulations that were too computationally taxing, we used a modified version of the algorithm BNReduction²³, which allows the identification of single stable state phenotypes that are most prevalent with our self-contained CASCADE topologies.

Model calibration by parameterization optimization

A genetic algorithm is applied to automate model parameterization, as follows:

The input for model calibration consists of:

²⁰ Kauffman, S. a. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3), 437–467. [http://doi.org/10.1016/0022-5193\(69\)90015-0](http://doi.org/10.1016/0022-5193(69)90015-0)

²¹ Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3), 563–585. [http://doi.org/10.1016/0022-5193\(74\)90172-6](http://doi.org/10.1016/0022-5193(74)90172-6)

²² Naldi, A. (2018). BioLQM: A Java toolkit for the manipulation and conversion of logical qualitative models of biological networks. *Frontiers in Physiology*, 9(NOV), 1–10. <http://doi.org/10.3389/fphys.2018.01605>

²³ Veliz-Cuba, A., Aguilar, B., Hinkelmann, F., & Laubenbacher, R. (2014). Steady state analysis of Boolean molecular network models via model reduction and computational algebra. *BMC Bioinformatics*, 15, 221. <http://doi.org/10.1186/1471-2105-15-221>

- Interactions: binary signed and directed interactions (SIF format²⁴).
- Steady state: Boolean vector containing states of nodes in input interactions, where nodes that should be active are assigned the value 1, and nodes that should be inactive are assigned the value 0. For nodes whose state cannot be determined a dash (-) can optionally be used to explicitly declare that the node state is undetermined. This steady state vector will be used for evaluating the correctness of a model's stable state.

The output from an automated model calibration is an ensemble of models with a stable state optimized to match the input steady state.

To run the parameter configuration, interactions are first assembled to logic equations, based on a default equation²⁵ relating a node with its regulators, for instance:

Target *= (A or B or C) and not (D or E or F),

where activating regulators A, B and C of a target are combined with logical ‘or’ operators, and inhibitory regulators D, E and F are combined with ‘and not’ operators to determine the state of the target node in the next time step. The operator ‘and not’, which directs the integration of activating and inactivating regulators, is referred to as the link operator in this manuscript. The topology is ‘self-contained’, meaning that any regulator is itself regulated by one or more components from within the network topology, effectively meaning there are no ‘user-controlled’ inputs to the network through e.g. hormone receptors.

Next, a genetic algorithm is used to iteratively refine the parameterization to produce a logical model with a stable state matching the specified input steady state. First, an initial generation of models is formulated, where a large number of mutations to the parameterization is introduced: randomly selected equations are mutated from “and not” to “or not”, or vice versa. For each model a fitness score is computed: each matching Boolean value between the vector of a stable state and the steady state improves the fitness score. Models without a stable state have a fitness of zero. Models with n stable states obtain a final fitness after integrating all Boolean values in a stable state vector and dividing the resulting overall fitness by n , thus penalizing models with multiple stable states.

For each generation, a user-defined number of models were selected for populating the next generation of models. For our simulations three models were selected, specifically the ones that achieved the highest fitness scores in each generation, to populate a next generation of 20 models. First, crossover was performed, where each selected model would exchange logic equations with other selected models (including itself, thus also enabling asexual reproduction). Then a number of mutations were introduced as described above. For our simulations up to

²⁴ Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., ... Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11), 2498–2504. <https://doi.org/10.1101/gr.1239303>

²⁵ Mendoza, L., & Xenarios, I. (2006). A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theoretical Biology and Medical Modelling*, 3, 1–18. <http://doi.org/10.1186/1742-4682-3-13>

three mutations were introduced. Before a stable state was obtained, the number of mutations introduced per generation was increased by a user-defined factor. The large number of mutations in the initial phase ensured that a large variation in parameterization could be explored. For our simulations we chose the factor to be 1000, effectively ensuring that the initial generations were randomly sampled from all possible model configurations. Evolution was halted when a user-specified threshold fitness was reached. In case this fitness could not be reached, evolution was halted when a user-defined maximum number of generations had been spanned. For our simulations we allowed for a maximum of 20 generations.

Model calibration by topology optimization

In order to introduce variations to the topology, the genetic algorithm modified a whitelist and a blacklist of regulators of the prior knowledge network, while always preserving at least one regulator for each target, so as to not break the self-contained property of the network. Based on the same formula as shown above,

Target *= (A or B or C) and not (D or E or F),

this means that the genetic algorithm takes out some subset of the regulators A, B, C, D, E or F (blacklisting). After a regulator has been eliminated, the genetic algorithm is also allowed to bring back regulators originally defined in the PKN (whitelisting). For our simulations, during the initialization phase we introduced 50 such topology mutations and when models with stable states were found, we reduced this number to 10, so as to not severely reduce the PKN edges.

Model simulation and synergy prediction

After repeating evolution a specified number of times, model ensembles were analyzed in a third step of the software pipeline, as follows:

The input to model simulation and synergy prediction consists of:

- An ensemble of logical models
- A drug panel: List of drugs and their target node(s) in the model
- Perturbations: the perturbations to be analyzed.
- Model output nodes with weighted score to evaluate global output (i.e. ‘growth’)

Output from model simulation and synergy prediction:

- Drug synergy predictions from ensembles of models

For each model, all perturbations specified were simulated. For each perturbation, the drug panel was consulted to fix the state of the specified node(s) to the value 0 (node state could in principle also be fixed to the value 1 for a drug that *activates* a signaling entity, but this feature was not used here as all drugs *inhibit* nodes in the model, thereby representing inhibition of their target in a cell). After simulating a perturbation, the global output parameter ‘growth’ was computed by integrating the weighted score depending on the states of model output nodes. For example, if two output nodes A (weight 1) and B (weight -1) were found to have the states A=1,

$B=1$ for a perturbation, the global output would evaluate to $A_{\text{state}} \times A_{\text{weight}} + B_{\text{state}} \times B_{\text{weight}} = 1 \times 1 + 1 \times (-1) = 0$. This value was then scaled from 0 to 1 based on the theoretical minimum and theoretical maximum ‘growth’, for this example, the range [-1,1], the global output would be 0.5. The scaled global output (‘growth’) was then used to compute synergies (see below).

All steps of the software pipeline were implemented in the OpenJDK Java v1.8 language and run on Linux 4.15.0-122-generic/Ubuntu 18.04.4 LTS. The pipeline can be accessed at <https://github.com/druglogics>.

In silico definition of synergy

Synergy is defined as an additional response beyond what is expected from a reference model of drug combination responses. Both for *in silico* simulations and *in vitro* experiments an observed combination effect can be formally defined as the effect E observed for two drugs a and b , where $E(a,b)$ is the observed effect in a combination experiment, $A(a,b)$ is the drug combination effect expected from each individual drug’s properties as based on a reference model for combination responses, and $S(a,b)$ is any difference between the observed and the expected drug combination effect, such that $E(a,b) = A(a,b) + S(a,b)$ ²⁶. In the case of excess effects observed for a combination, $S(a,b)$ is positive and synergy is called, and conversely for attenuated effects, $S(a,b)$ is negative and antagonism is called. Finally, for drug combinations where $E(a,b)$ equals $A(a,b)$, the drug combination effect can fully be anticipated by each drug response independently, and neither synergy nor antagonism is called.

In model simulations the expected drug combination response is defined as the product of the two global output ‘growth’ values for each single drug, similarly to the Bliss independence²⁷ synergy metric used in lab experiments: when a combinatorial perturbation in simulations is found to predict a lower growth than expected, i.e. $\text{growth}(a,b) < \text{growth}(a) * \text{growth}(b)$, the combinatorial perturbation response is declared synergistic.

Gold standard synergies

Our previously published dataset of targeted drug combinations^[17] was used to benchmark the algorithms. The drugs included comprised the inhibitors 5Z-7-oxozeanol (5Z), AKTi-1,2 (AK), BIRB0796 (BI), CT99021 (CT), PD0325901 (PD), PI103 (PI), PKF118-310 (PK), JNK Inhibitor XVI (JN), BI-D1870 (D1), BI605906 (BIX02514) (60), Ruxolitinib (INCB18424) (SB), SB-505124 (RU), D4476 (D4), KU-55933 (KU), 10058-F4 (F4), Stattic (ST), GSK2334470 (G2), GSK-429286 (G4), P 505-15 (P5). For the drug synergy calling in the 153 combinations drug screen, three curators were asked to evaluate growth curves and decide on which showed

²⁶ Li, H., Li, T., Quang, D., & Guan, Y. (2018). Network propagation predicts drug synergy in cancers. *Cancer Research*, 78(18), 5446–5457. <https://doi.org/10.1158/0008-5472.CAN-18-0740>

²⁷ Bliss, C. I. (1939). The Toxicity of Poisons Applied Jointly. *Annals of Applied Biology*, 26(3), 585–615. <http://doi.org/10.1111/j.1744-7348.1939.tb06990.x>

interesting combination effects that could have warranted further investigations. A consensus list was then used to identify a threshold for drug synergy assessment using the software Clmbinator²⁸ and configured to compute synergies per the Bliss metric. The analysis identified six drug synergies (AK-BI, PI-D1, BI-D1, PI-G2, PD-PI, 5Z-PI). Note that two drug synergies in the drug screen performed in 2015 were not captured by this analysis, probably relating to the different readouts used in the drug screen performed in 2019 (xCELLigence and CellTiter Glo, respectively).

Normalization

Normalization of synergy predictions was performed by computing the exponential fold change for the ratio of output from models *calibrated* to steady state biomarker data (x) to output from models calibrated to a *random* yet proliferative phenotype (y):

$$\text{synergy} = \exp((\text{growth}_x(a,b) - \text{growth}_x(a) * \text{growth}_x(b)) - (\text{growth}_y(a,b) - \text{growth}_y(a) * \text{growth}_y(b)))$$

Our random proliferative phenotype corresponds to a cell with all anti-apoptotic signals inactivated, and at least one prosurvival signal active.

Mouse xenograft experiments

40 female Balb/c mice 4-5 weeks old (Taconic) were inoculated with two million AGS cells subcutaneously in the right dorsal flank. Cells were mixed with Matrigel to improve probability of successfully establishing a xenograft model: in a small pilot (n=3) we observed that none of three mice injected with cells in medium (DMEM) developed tumors, while two of three mice injected with cells in medium and Matrigel developed tumors. 100 µl of cell suspension in HAM'S F12 medium (Invitrogen, Carlsbad, CA) with 10% fetal calf serum (FCS; Euroclone, Devon, UK), and 10 U/ml penicillin-streptomycin (Invitrogen) was mixed with 100 µl of ECM Gel from Engelbreth-Holm-Swarm murine sarcoma (Sigma-Aldrich). After four weeks, minuscule but palpable tumors had formed in 30 mice, which were randomized to four groups and subjected to treatment: 1) 5Z-7-oxozeaenol (3 mg/kg/d), 2) PI103 (5 mg/kg/d), 3) 5Z-7-oxozeaenol (3 mg/kg/d) + PI103 (5 mg/kg/d), 4) vehicle. Randomization ensured that average tumor volume was similar in the four groups. Weights of mice ranged from 14.9 to 20.0 grams at onset of treatment, with average weight 17.66 g and standard deviation 1.06 g. All mice received the same dose of drugs, and the dose was adjusted for a body weight one standard deviation below average, i.e. 16.6 g. Drugs were diluted in medium with 40% DMSO for a total injection volume of 250 µl and injected intraperitoneally three times per week for a total of seven injections. Maximum (a) and minimum (b) tumor diameters were measured twice weekly with a caliper, and the volume V of the tumor was estimated from the formula $V = 0.5 a \times b^2$.

²⁸ Flobak, Å., Vazquez, M., Lægreid, A., & Valencia, A. (2017). Clmbinator: A web-based tool for drug synergy analysis in small- and large-scale datasets. Bioinformatics (Oxford, England), (March), 1–3. <https://doi.org/10.1093/bioinformatics/btx161>

Results

For reliable simulation of drug responses of cancer cell lines, computational models must adequately represent the regulatory network (topology) underlying cell fate decisions, meaning that high quality molecular causal relationship data must exist and be converted to regulatory graphs. In addition, the activity states of molecular regulatory components must be measured, demanding high quality biomarker data. From the regulatory graph the response of components to upstream source nodes and influence on downstream target nodes needs to be specified in the form of logical rules and calibrated so as to accurately represent the biological decision mechanisms of these cells in a Boolean framework. Finally, good benchmarking data must exist to evaluate the performance of the model, e.g. for our purposes in the form of drug synergy data, see figure 1A.

Design of an automated model parameterization module

Previously we have shown the feasibility of logical model predictions of drug synergies¹⁷ using the cancer cell line AGS, chosen due to known deregulations of several core cell survival signaling pathways. A Prior Knowledge Network (PKN) was curated to represent these signaling pathways, and converted to a set of mathematical rules formulated in Boolean logic. This model, available from <https://github.com/druglogics/cascade> as CASCADE 1.0 (Fig. 2A), consists of 75 nodes representing cancer signaling entities and 149 edges representing regulatory interactions, and it could predict five synergies of which four were experimentally confirmed¹⁷. Since our model is based on prior knowledge, amenable to interpretation by molecular biologists, the model also can be used for inspection as to which signaling pathways are important for particular drug response observations, e.g. we have suggested that FOXO signaling was crucial to the drug synergy effect of joint PI3K and MEK inhibition¹⁷. Whereas for many of the logical rules the definition of the logical operators (AND, OR, NOT) was more or less evident from literature and database knowledge, analysis of Boolean model attractors indicated that some logical rules needed further manual optimization in a stepwise manner so that ultimately the model stable state behavior matched the observed pattern of signaling entities at steady state in proliferating AGS cells. We now report on how we automated and generalized these steps required to parameterize an *in silico* model of a cancer cell line, by employing a genetic algorithm for deriving logical rules from prior knowledge and steady state signaling observations, see figure 1B. From a curated network topology a set of standardized logic equations are obtained by defining one logic equation for each model target node, with model source nodes as operands²⁵. For example, if protein T is activated by proteins A and B, while protein C inhibits protein T, the equation could read as $T = (A \text{ OR } B) \text{ AND NOT } C$. Subsequently, the parameterization (choices of logical operators) is optimized by a genetic algorithm, specifically modifying the AND NOT/OR NOT parameter. The genetic algorithm iteratively modifies the parameterization of a small subset of the equations, and selects best performing models for defining a new generation of candidate models. Best performing models are chosen based on their compliance with reproducing known baseline cell signaling states, as far as the available cell line data allows it. Evaluating fitness from a match with baseline

observations also means that our models are defined independently of perturbation data. Our software solution for automatic parameterization is available at <http://github.com/druglogics/>.

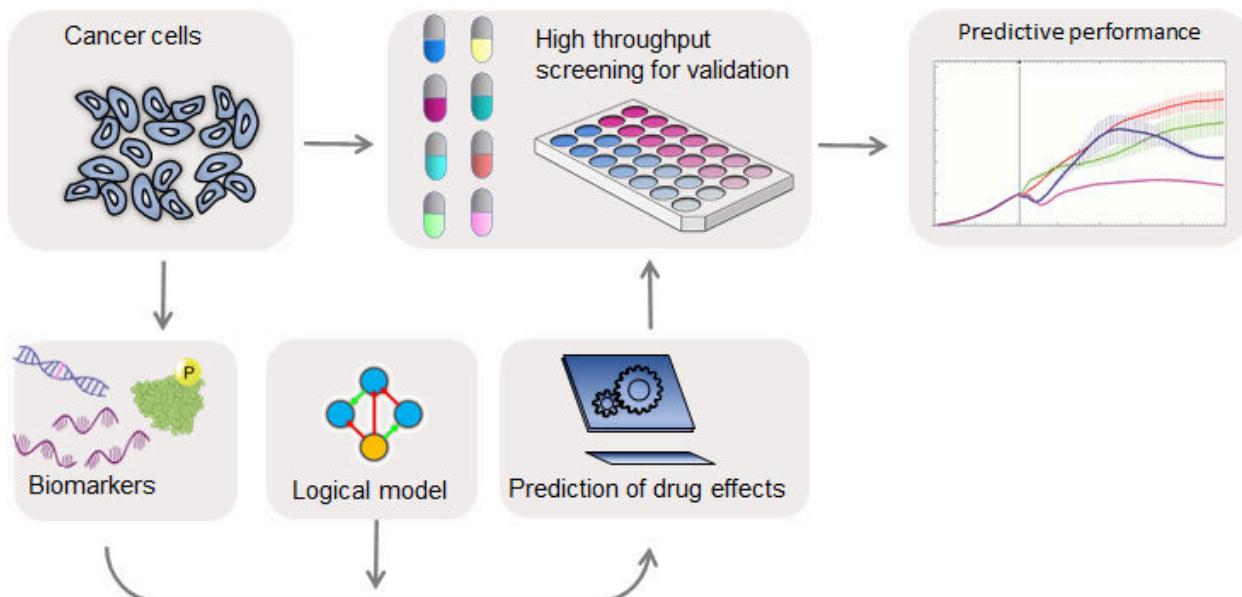


Figure 1A: Overview of the drug synergy prediction platform. Cancer cells are analyzed for biomarkers used to define logical models that can be used to predict drug synergies. Model predictions are tested by benchmarking against high throughput drug screens.

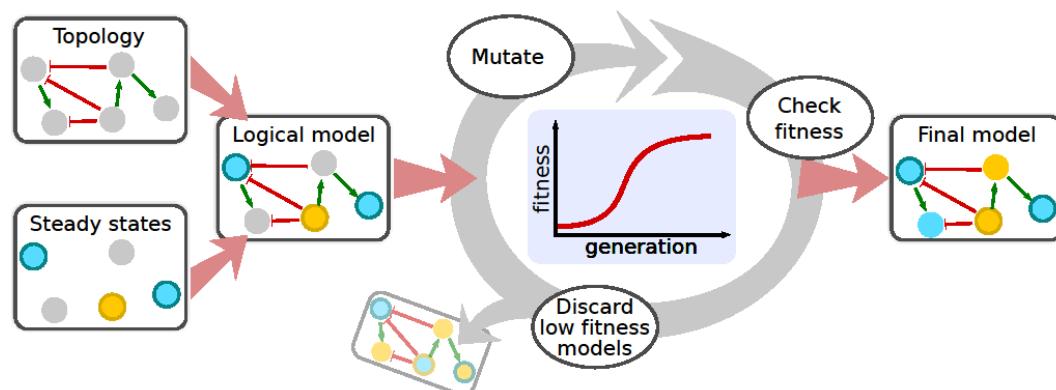


Figure 1B: A genetic algorithm optimizes logical models to cancer cells. A prior knowledge signaling topology representing molecular causal interactions is taken as input to define logical models with predefined rules as initial logic equations. A genetic algorithm will iteratively randomly choose logical rules by mutating the AND/OR configuration, thereby re-parameterizing a logical model until a model shows a maximum compliance with steady state signaling observations (biomarkers). This procedure is performed in parallel for hundreds of models until an optimized ensemble of models is available for drug synergy prediction.

Automated logical rule definitions perform on par with manually curated rules

To compare results from our manually constructed logical rules with automated rule definitions, our software translates the graph, as encoded in a SIF file format, to a set of 75 logic equations in a standardized format²⁵. Logic equations are then optimized using a genetic algorithm in a process where the fitness of each model is calculated by comparing the matches of its stable state nodes with observed activities of signaling entities for proliferating cancer cells. For the AGS cells, this process comprised 20 generations, in which each generation received mutations to a small subset of logic equations iteratively, with 20 models per generation tested for fitness. In order to adequately cover the space of local optima, the evolutionary process was repeated 50 times and the three best performing models from each evolution were retained, which resulted in an ensemble of 150 models. A theoretical maximal fitness of one would be reached if all nodes have a state matching the observed activity state of the corresponding protein. As can be seen in figure 2B, the population average fitness of each generation increases exponentially before plateauing at a fitness close to the theoretical maximal fitness, per Holland's Schema Theorem²⁹, indicating that the theoretical models can be parameterized so as to be compliant with experimentally observed signaling states. While a genetic algorithm cannot guarantee a global optimum, our results clearly indicate that we quickly achieve convergence to a local optimum.

Whereas these model ensembles provide the testing ground for the *in silico* drug effect simulations, it is to be expected that certain motifs of the network topology itself will create ‘blind spots’ resulting in some synergies to be impossible to predict. For example, if two directly sequential signaling nodes are targeted by two different drugs, while no other influences from other signaling entities are allowed by the topology, then these two drugs cannot be predicted to act synergistically in our logical modeling framework. To remedy such situations extensions to the prior knowledge network are necessary, or conversion to non-discrete modeling. On the other hand, if two drug targets are active and are the only (activating) source nodes of a joint downstream target node, with their joint effect on the target governed by an OR logical operator, this may constitute a synergy that is highly likely to stand out in an analysis, since the OR operator would cause either drug alone to not affect a joint downstream node. Between these two extremes, the topology will be more or less likely to produce a particular synergy prediction for a given combination perturbation. In order to correct for topology-intrinsic propensities for predicting some synergies we next employed a normalization strategy where synergy predictions for an automatically parameterized model ensemble are normalized to a *randomly* parameterized model, meaning a model ensemble that covers many different selections of OR and AND operators, irrespective of any particular stable state. This means that in our further analyses we used the fold change of the predicted synergy score of a test model against a randomly yet proliferative parameterized model (see Methods).

²⁹ Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. Adaptation in Natural and Artificial Systems (Reprint). Cambridge, Massachusetts: The MIT Press.
<http://doi.org/10.7551/mitpress/1090.001.0001>

We first tested our software pipeline by considering predictions from model ensembles for simulating the 21 drug combinations that were analyzed previously¹⁷. Synergies were defined as a predicted ‘growth’ output for two drugs together being lower than the product of each individual drug’s ‘growth’, analogous to the Bliss synergy metric³⁰ used in cell culture lab experiments (see Methods). Among 21 drug pairs, 15 were predicted to act synergistically by this definition, exhibiting a range of synergy strengths, and quantified performance of these models was surprisingly high: by selecting different thresholds for synergy predictions a receiver-operating characteristic (ROC) curve (sensitivity vs 1-specificity) shows a ROC area-under-curve (AUC) of 0.97 and a precision-recall (PR) AUC of 0.91, see figure 2C. The analysis shows that the top six predictions comprise all four experimentally validated synergies. Notably, the automatically parameterized models produced no false negatives. This effectively means we could in principle have reduced our full drug screen to only test 29% percent of the combinations (6 out of 21), had we guided our experiments by model simulations, which is comparable to the performance in our manually parameterized model¹⁷.

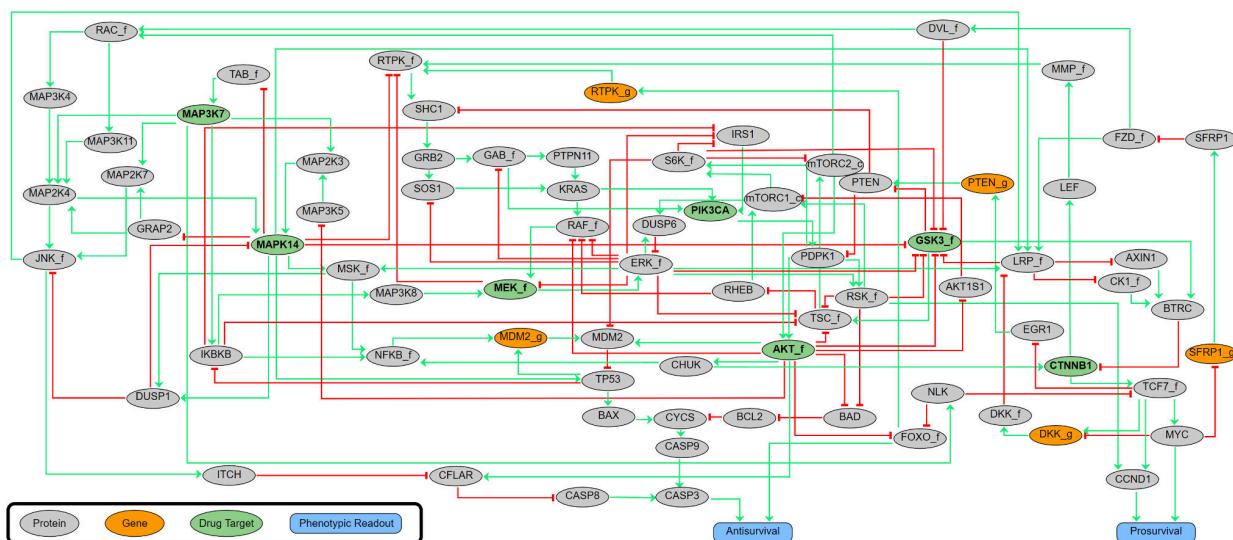


Figure 2A: The CASCADE 1.0 prior knowledge network. 75 signaling nodes with signed and directed regulatory influences annotated (activating interactions in green, inhibiting interactions in red). All signaling components receive input from other signaling components from within the network, and ultimately influence the two phenotypic output nodes Antisurvival and Prosurvival.

³⁰ Bliss, C. I. (1939). The Toxicity of Poisons Applied Jointly. Annals of Applied Biology, 26(3), 585–615. <https://doi.org/10.1111/j.1744-7348.1939.tb06990.x>

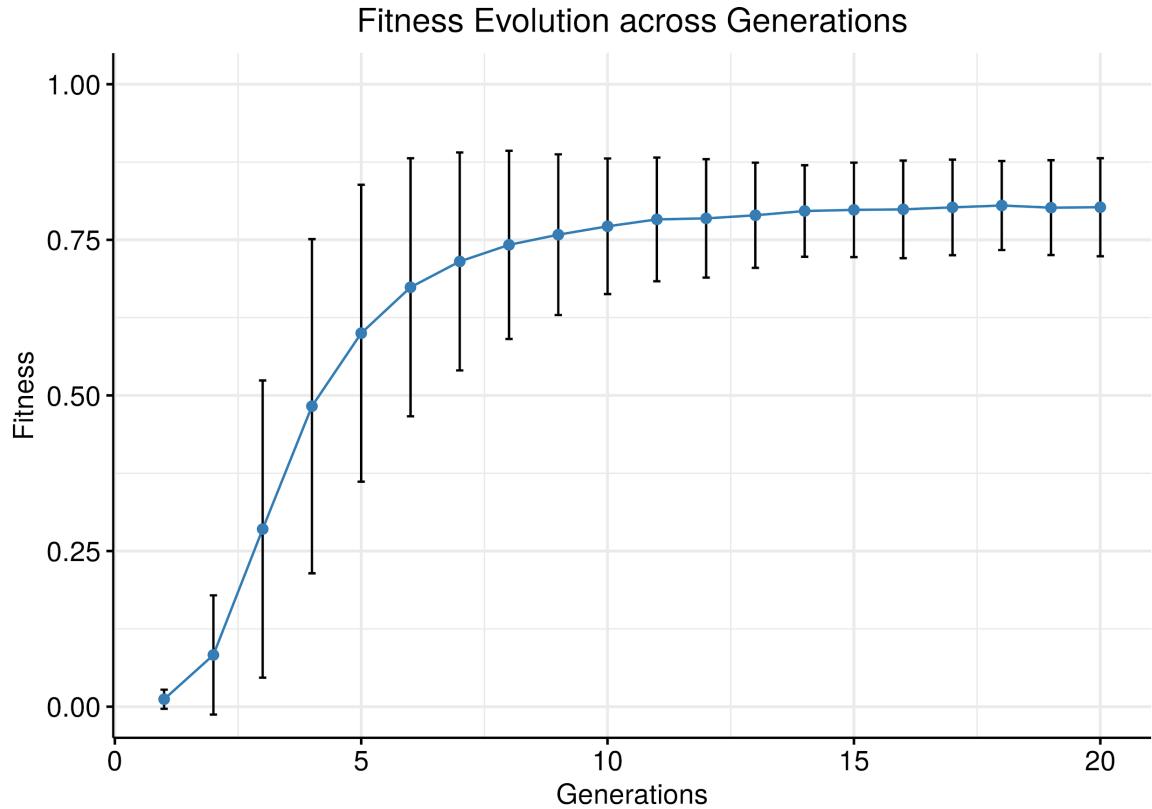


Figure 2B: Evolution of fitness of calibrated models. Overall fitness is plotted as a function of generation, with average fitness and standard deviation indicated. The data for this figure was produced by running the genetic algorithm for 1000 simulations, with 20 generations per simulation and 20 models per generation. We observe that the average fitness and standard deviation follow a sigmoidal increase and stabilize after 10-15 generations. The persistence of the standard deviation across generations including those late in the evolution shows that new models still explore variations to the model parametrization while selection keeps the fitness score of the trained models at a constant plateau.

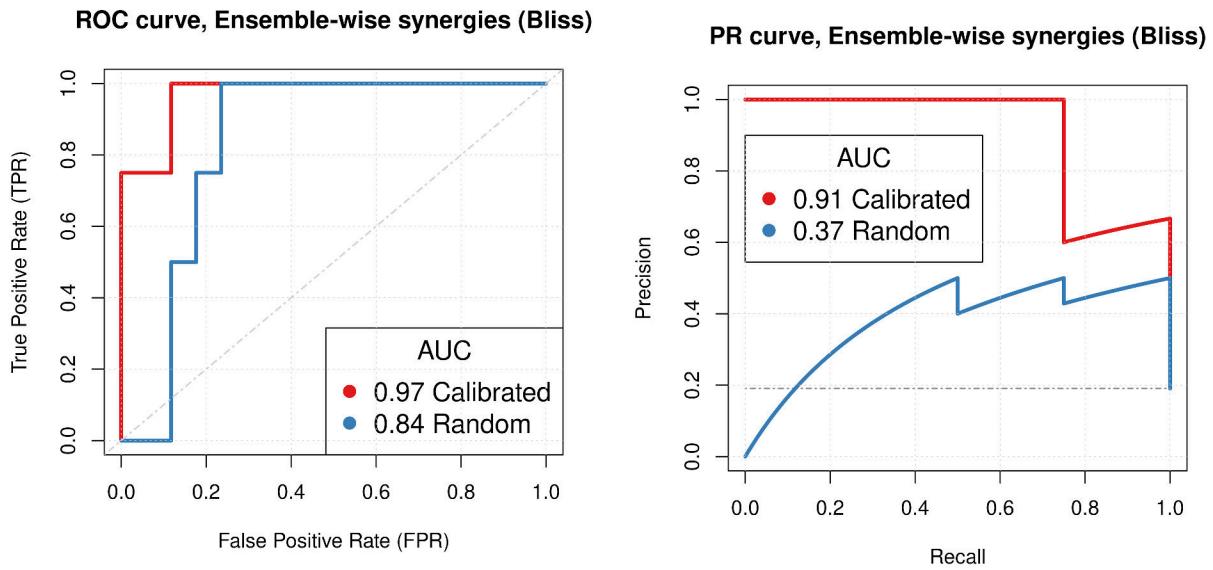


Figure 2C: Predictive performance of ensembles of logical models. ROC curves are in the left panel, and PR curves are in the right panel. Random model predictions were generated by collecting predictions from ensembles of models trained to a random yet proliferative phenotype. Calibrated predictions were generated by model ensembles, trained to steady state data, and normalized to the random model predictions (see Methods for more details). The genetic algorithm modified the balance of influence between positive and negative regulators of a target node, while topology features (edges, nodes) were not modified. Both ROC and PR curves show very good performance across all model sets for the calibrated models, similar to results from Flobak et al (2015).

Automated model optimization as a solution for larger model topologies

The benefits of automatic parameterization become more apparent in calibration of models with larger topologies. We demonstrate this with the CASCADE 2.0 model, which is a manually curated cancer signaling topology comprising 144 nodes and 367 edges. CASCADE 2.0 includes pathways with TGF-beta, JAK-STAT, and Rho GTPases, as well as extensions of pathways already present in CASCADE 1.0, to enable simulation of a larger set of drug combinations¹⁸. Starting with this large curated model, we analysed in more detail the effects of automated model training while randomly mutating logical rule configurations and network connectivity, and assessed the results against the biological regulatory mechanisms that were affected. We varied several aspects in the training protocol, each time assessing the effect on the performance of the models for correct synergy prediction:

- Optimising logical rules against partially incorrect calibration data
- Optimising the regulatory network by stepwise, random removal/inclusion of edges
- Checking the effect of random rewiring of the regulatory network

For each of these model alteration strategies, we not only looked for overall fitness but also in more detail at the represented biological mechanisms that were affected, to judge whether improved or reduced simulation performance could be reconciled with involvement of proteins of regulatory interactions in the context of cancer. The hypothesis was that, while taking the overall value of curated prior knowledge as a given, the relevance of individual regulatory interactions and the precise mathematical representation of their regulatory effects in specific cancer environments might be difficult to infer from papers and therefore could be algorithmically improved. The effects of the network connectivity and rule mutations were judged in model ensembles and compared with observed synergies. For each model able to reach a stable state, mutations also allowed to assess mutual dependencies between edges or subsets of edges and corresponding rules, possibly indicating context dependence. This allowed us to identify parameters and edges that appeared to be essentially fixed and thereby of fundamental importance for model performance.

We compared simulation results from automatically trained ensembles of 450 models to drug synergies in a new drug screen comprising 153 combinations of 18 targeted drugs³¹. We found that, in contrast to the CASCADE 1.0 predictions, normalization of topology-intrinsic prediction propensities was critical to the predictive performance (see Supplementary Figures 1 and 2). We find that models obtained by automated optimization, as described above, could predict drug synergies with a ROC AUC of 0.69 and a PR AUC of 0.18, clearly outperforming random model predictions (see Figure 3). This means that our drug screen could have been reduced from blindly testing all 153 combinations to only the screening of 36 combinations, which would increase the synergy prevalence of tested combinations from 4% (6 of 153) to 11% (4 of 36). We would dismiss 117 drug combinations but at the expense of missing two observed synergies.

³¹ Flobak 2019

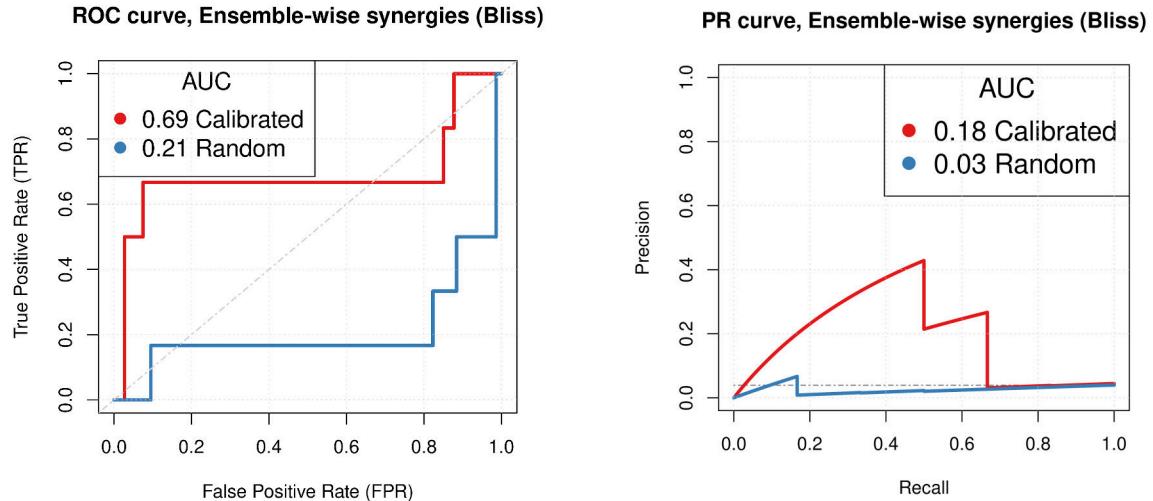


Figure 3: Model performance. Predictive performance of calibrated and random models based on the CASCADE 2.0 model was tested against data from a corresponding drug screen³¹. Models have logical rule mutations only and Bliss Independence was used to assess the model performance. We observe that correctly calibrated models perform substantially better than random models.

Topology and calibration data needs to be correct

From here on we focus on the CASCADE 2.0 topology, for additional analyses see Supplementary Material, which has similar experiments with the CASCADE 1.0 topology, underpinning conclusions similar to those drawn here.

Impact of data quality on model performance

Since our models are derived from prior knowledge and calibrated based on sample-specific measurements (calibration data), modifications to both the prior knowledge and data must be expected to affect the predictive performance of the models. We first checked how the quality of the calibration data affected model predictions. The performance of models trained to partially incorrect calibration data was expressed as PR AUC and, when plotted against the fitness of these models compared to the fully correct calibration data, we observe that higher PR AUC correlates with higher fitness of models, indicating that calibration of models indeed improved synergy predictions for our dataset (see figure 4). However, even models trained to highly incorrect data, with roughly 50% of calibration data flipped (meaning a true fitness around 0.5), perform better than a random classifier (PR AUC 0.04), indicating that model topology alone already carries information that can be leveraged to predict drug synergies. Note that due to the stochasticity of model calibration, the model ensemble average fitness never reached the extreme fitnesses below ~0.3.

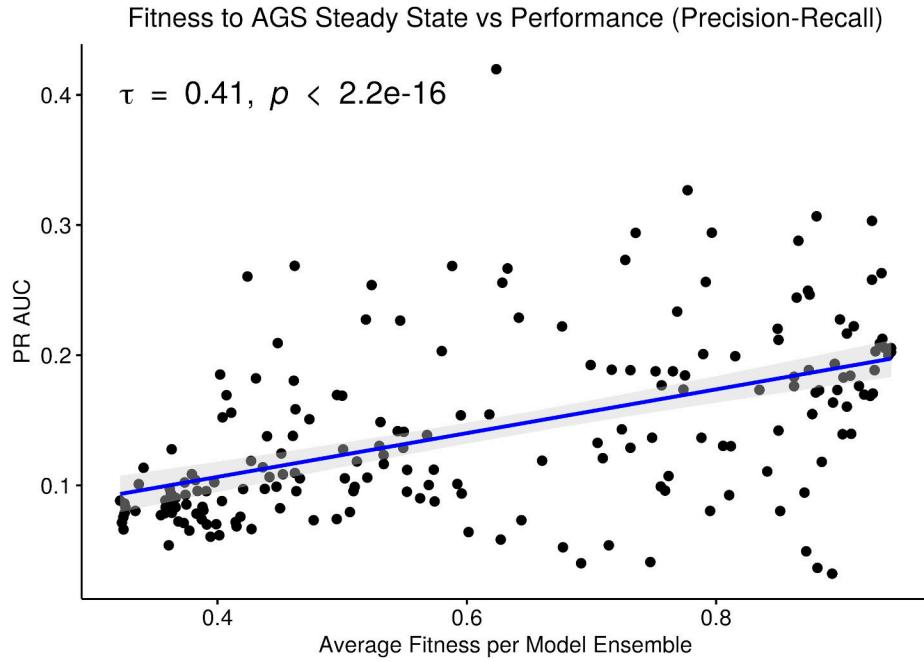


Figure 4: PR AUC performance dependence on fitness. Each model ensemble, displayed as one dot in the scatterplot, was trained to a partially incorrect steady state signaling profile derived from the biological phenotype of the AGS cell line¹⁷. A total of 205 training profiles were created, each one used to generate one model ensemble consisting of 60 models. The x-axis reports the average fitness of each model ensemble as evaluated to the curated steady state. Because of the non-normality of the data, the Kendall rank-based correlation³² test is used to derive the proposed association.

Randomizing regulatory edges of the curated model reduces predictiveness

As the quality of the calibration data does impact model performance, but not obviate it even if these data are highly incorrect, we next explored the value of the quality of the curated regulatory graph topology. We generated a series of models with various degrees of ‘scrambled’ topologies with modified causal interactions in the regulatory graph (of the type: source - effect - target), by randomly exchanging a particular ‘source’ in a causal interaction with the source of another interaction and investigated the performance of models with these incorrect regulatory interactions. Similarly, we investigated the impact of randomly reassigning target nodes, as well as the impact of inverting signed effects, i.e. from inhibition to activation, and vice versa. Note that we will later explore the effect of missing prior knowledge (simple deletions), while here we present results for incorrect prior knowledge. The results (Figure 5) show that even low levels of randomization in the curated knowledge significantly reduce the predictive power of the models, quickly approaching random performance. Overall, we conclude that both calibration data and

³² Kendall, M.G. (1948). Rank correlation methods. Griffin

prior knowledge quality are important to correctly predict drug synergies, and that errors can be detrimental.

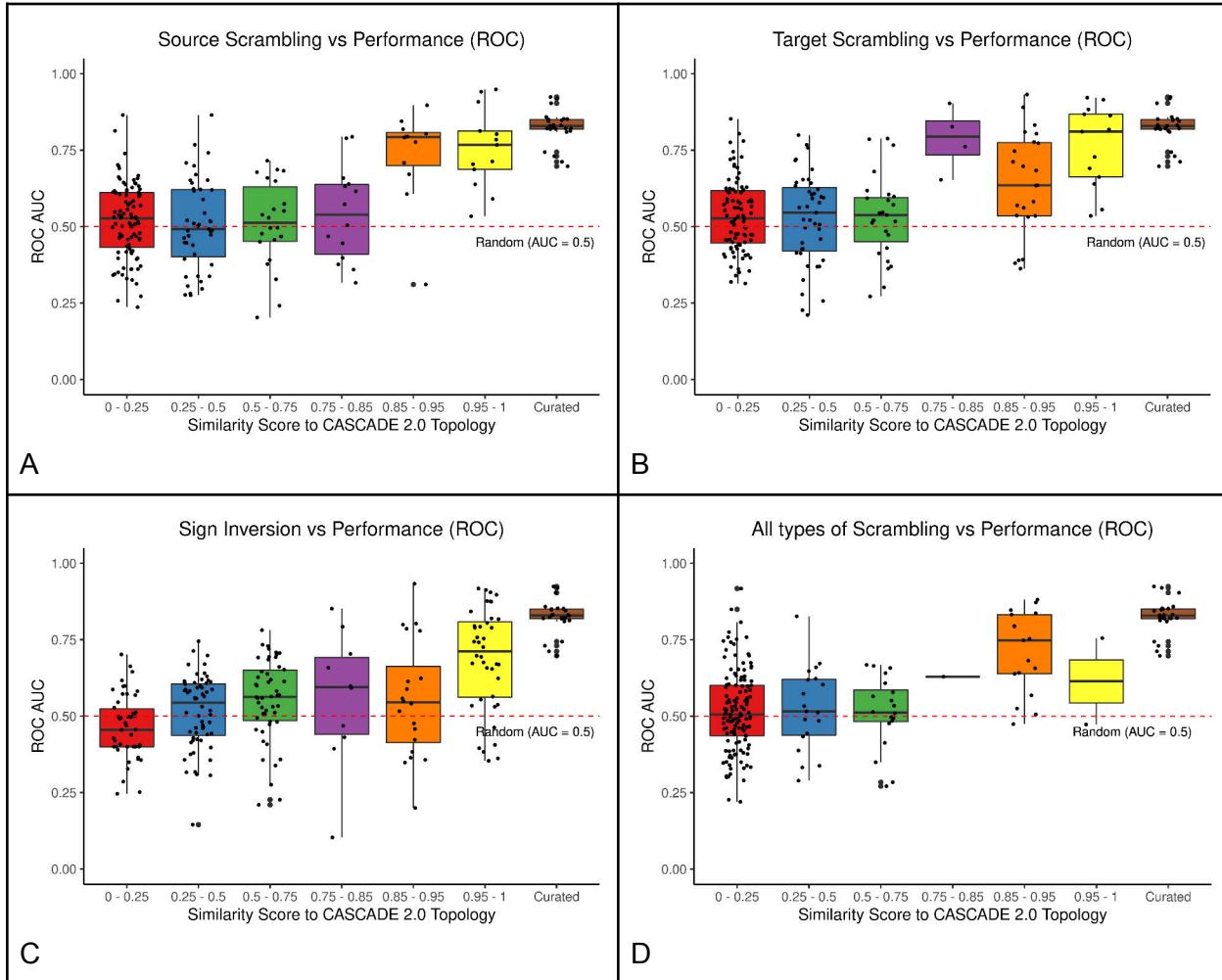


Figure 5: Effects of variations introduced in the CASCADE 2.0 prior knowledge graph. Panel A shows the effect of reassigning source nodes in causal interactions, panel B shows the effect of randomly reassigning target nodes in causal interactions, panel C shows the effect of randomly inverting activation/inhibition annotation, and panel D shows the result for all types of modifications introduced simultaneously. Each box plot shows a graded response for the predictive performance from complete modifications (left) to less substantial modifications (right). Each dot represents a different model ensemble generated from the associated topology, calibrated to the AGS steady state, and normalized to a random yet proliferative profile. The “Curated” group refers to model ensembles bootstrapped from a pool of models generated using our optimization algorithm from the original CASCADE 2.0 topology. See Supplementary Material Figure S4 for a similar analysis with precision-recall as performance metric, and Figures S5 and S6 for the same analysis done on the CASCADE 1.0 topology.

Model ensemble heterogeneity and mechanistic insight

To appreciate the heterogeneity amongst models in model ensembles obtained through the parameter optimization, we studied both attractor and parameterization heterogeneity against model fitness, in subsets of these ensembles selected for specific features (model sub-ensembles). In the heatmap grouped by K-means clustering (Figure 6), calibrated models to a relatively large extent obey the calibration data, with states of steady state nodes mostly identical to the data to which they were trained (the subset of nodes (24 of 144) that were specified in the calibration data). Model stable state vectors (rows in Figure 6) have notable areas of homogeneity, as judged by large stretches of nodes (indicated on the X-axis) that are either all activated (green) or inhibited (red) in all models, but in other areas (e.g. the upper-middle panel of the heatmap) the heterogeneity and discrepancies with calibration data is quite substantial. This heterogeneity was much more widespread in the parameterization space. For some nodes there is high correlation between their parameterization (link operator AND-NOT vs OR-NOT) and stable state (Inactive or Active, respectively), but for many the correlation is surprisingly low (see Agreement panel in Figure 6). These observations indicate that a) a limited set of training nodes was sufficient to provide homogeneity in parts of the attractor space, and b) large heterogeneity in the parameterization space still can be compliant with homogeneity in the attractor space (see in particular the large green (active) area of the stable-state heatmap). In other words: there are many logical rule configurations that yield models properly representing biological observations compliant with calibration data. This underpins the decision to use model ensembles rather than single models, since these ensembles cover a larger set of parameterizations (behavior) that are all compliant with the input data.

The analysis of node states and parameterization allows us to investigate mechanisms underlying observed behavior and to look for biological explanations for some of the observations. As shown in Figure 6, indicated by panel ‘COSMIC’, the models allow the prediction of activity of several proteins implicated in cancer. Figure 7 shows the analysis of their activity state, and it appears that proteins from genes annotated as oncogenes in COSMIC tend to be active, while proteins from genes annotated as tumor suppressor genes (TSG) tend to be inactive in overall steady states. This is biologically plausible and attests to the capability of our mechanistic model to generate hypotheses about the underlying molecular biology.

The training of the models to biomarker data never results in the absolute maximum fitness (1), and the stable state analysis (Figure 6) shows that three data points are most often violated: JNK signaling (JNK_f), ERK signaling (ERK_f) and p38 signaling (MAPK14), all member of the MAPK pathway (see Figure six, stable state panel, black rectangles). These network nodes are all clustered in the highly heterogeneous section in the steady state heatmap. In the manual curation of the CASCADE 1.0 topology¹⁷ it was noted that reports on the activity of ERK in AGS cells varied frequently, with only slightly more than half of the publications reporting ERK to be active. We found that the predictive performance of model versions with ERK being active was significantly higher than the sub-ensemble where ERK is inactive (see Figure 8), suggesting that from a functional point of view ERK should be considered active in AGS cells.

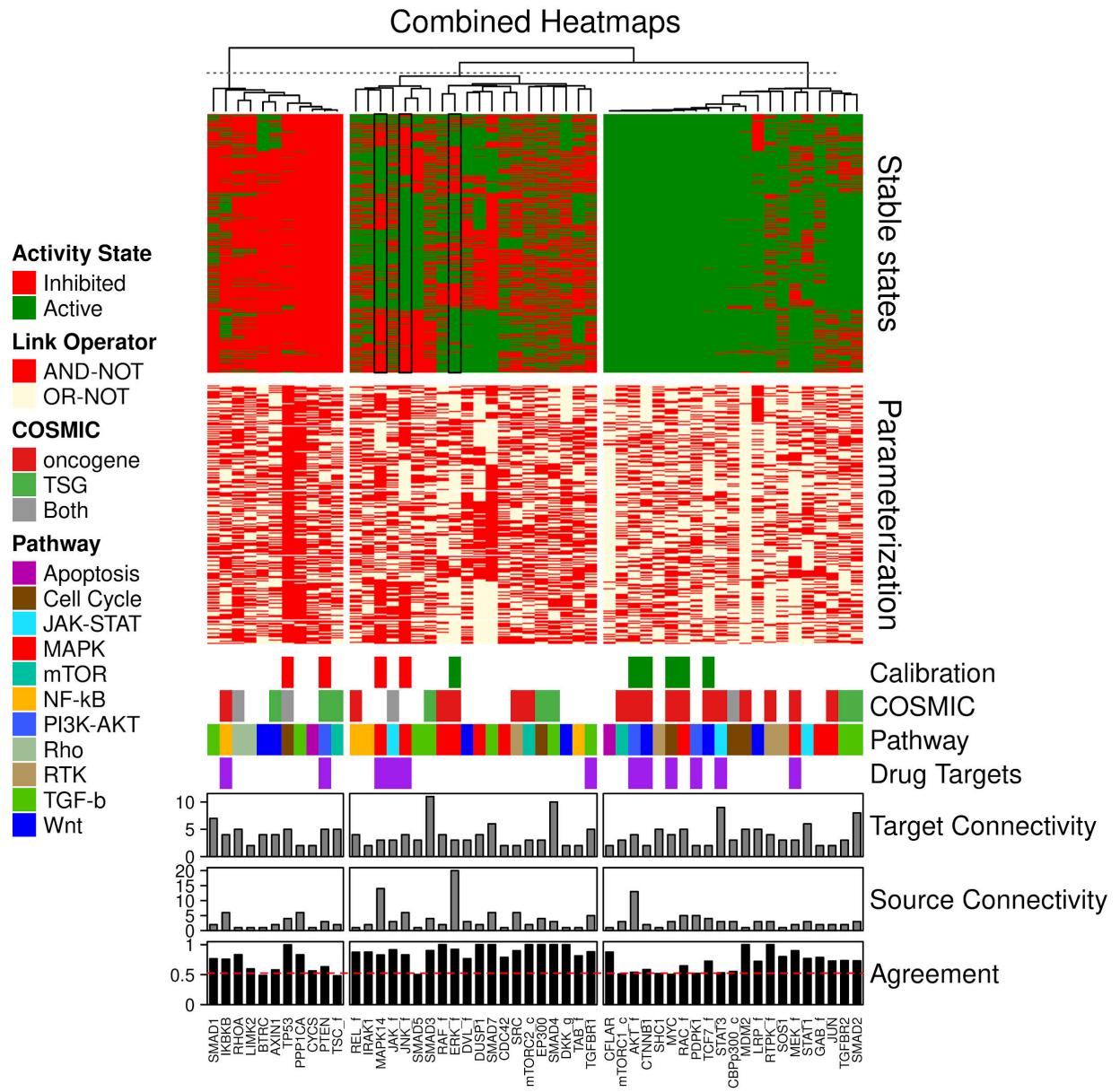


Figure 6: Combined stable states and parameterization heatmaps. A total of 4500 Boolean models were used for this analysis. Only the CASCADE 2.0 nodes that have a link-operator in their respective Boolean equation are shown. The 52 link-operator nodes have been assigned to 3 clusters with K-means using the stable states matrix data. The link-operator data heatmap has the same row order as the stable states heatmap. Steady state data (Calibration), COSMIC classification of tumor suppressor genes (TSG) and oncogenes, pathway association (Pathway), drug target characteristic (Drug Targets), in-degree connectivity (Target connectivity), out-degree connectivity (Source connectivity) and percent agreement between parameterization and stable state annotations (Agreement) are indicated below the heatmaps.

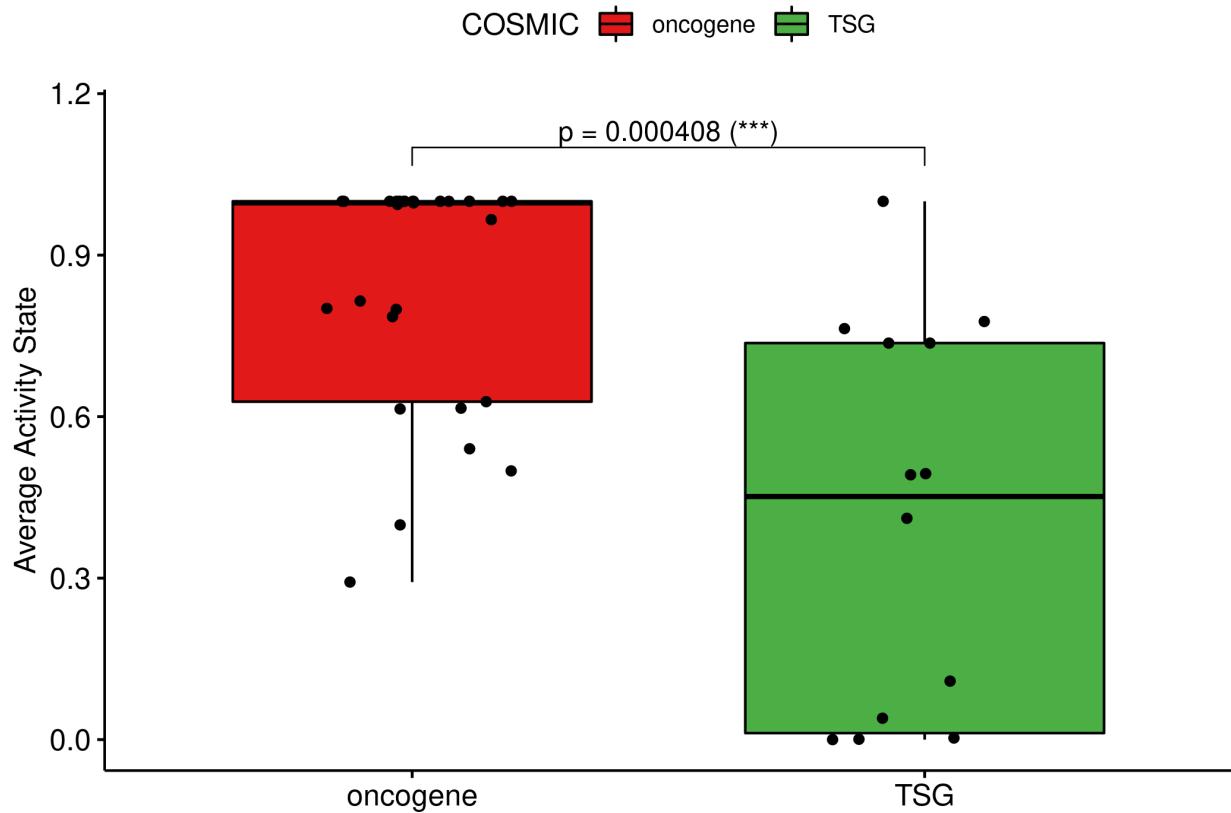


Figure 7: Box-plot of stable state protein activities. The stable state models yield activity values for all proteins and these activities are displayed for oncogene and tumour suppressor gene proteins. Proteins from oncogenes (left) tend to be designated as active, while proteins from tumor suppressor genes (TSG, right) tend to be designated inactive.

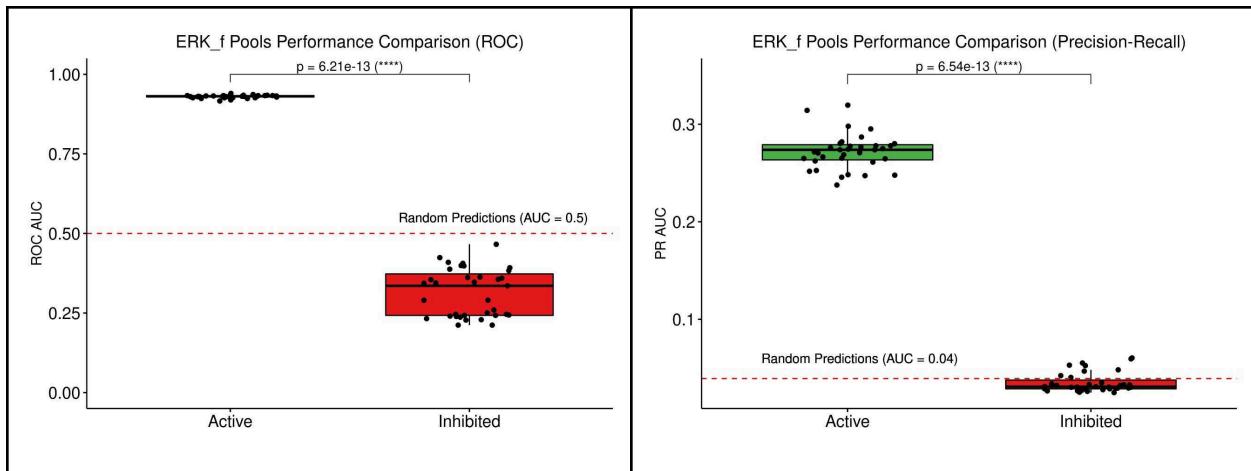


Figure 8: Comparison of predictive performance of model sub-ensembles. The model set with ERK active scores better than the models with ERK inactive, both for ROC (left panel) and PR (right panel).

Rule and edge optimization identifies key regulatory mechanisms

Traditionally, logical model definitions start out with a prior knowledge graph, after which a most optimal parameterization is sought based on experimental evidence and model behavior. We asked which was most influential to accurately predict synergies: alterations to the topology or to the parameterization, in the evolution to maximum fitness. We configured the genetic algorithm to modify edges in the topology, by either removing or subsequently restoring edges from the initial prior knowledge network (as long as no signaling component lost all its source inputs), or to modify the parameterization of logical rules. We found that modifications of the edges and the parameterization both resulted in substantially improved prediction performance, significantly better than a random classifier. In particular, the performance (as evaluated by precision-recall) was very high for topology-mutated models, even outperforming models trained by parameterization optimization. While ROC AUC was consistently high around 0.8, the PR curve benefited by displaying very high positive predictive values at very conservative sensitivity thresholds, meaning that a predicted drug synergy is highly likely to represent an actual synergy, see Figure 9. Note that there is a major difference between missing data and incorrect data: as was previously demonstrated (Figure 5), model predictive performance suffered severely from included and incorrect prior knowledge, while model predictive performance can improve by omitting putatively correct prior knowledge.

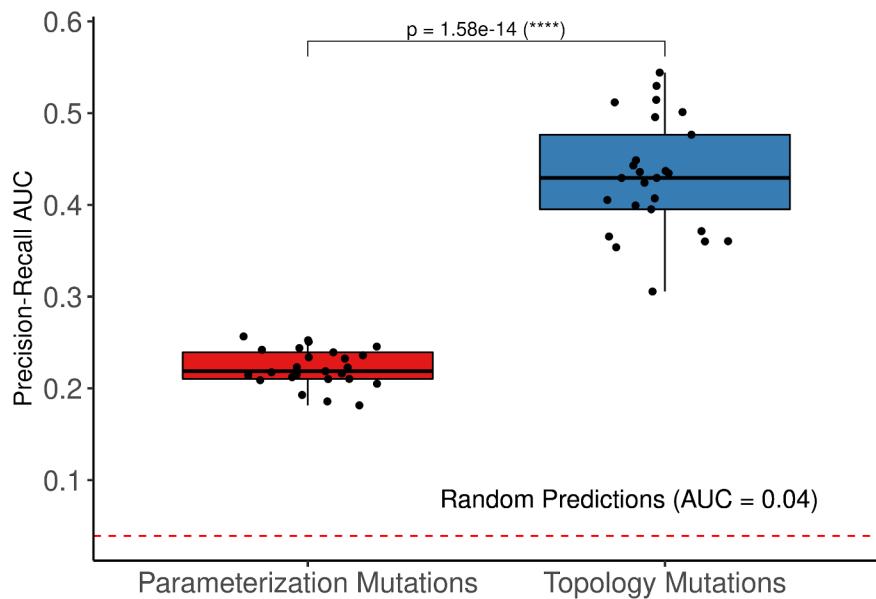


Figure 9: Model performance after parameter and topology modifications. Mutating parameterization (left) and topology (right) both tend to improve synergy prediction performance, as evaluated by precision-recall AUC. Models with topology modifications perform better than models with parameterization modifications.

Looking at the topology modifications, we hypothesized that deletion of certain edges could be favored by the genetic algorithm, to obtain maximum fitness. Every node in CASCADE 2.0 is annotated to a specific pathway¹⁸, allowing us to assign all edges to a specific pathway, if both source and target node belong to the same pathway, or to crosstalk for edges that link nodes from different pathways, see figure 10. We observed that certain edge groups are always preserved (the left-most cluster), while other edges are very likely to be removed (cluster on the right). Interestingly, a majority of these removed edges belong to the TGF-beta pathway, in particular representing inhibitory effects of the protein SKI and other inhibitors of SMADs. In the model, SKI itself is inhibited by active AKT signaling, and thus removal of inhibitory edges from SKI allows restricting the activity of some of the SMAD proteins in the model, in particular to SMAD1, SMAD3 and SMAD4, which tend to be inactivated in the topology mutated models. It is evident from figure 10 that crosstalk is largely preserved during model optimization, potentially relating not only to sparse knowledge of crosstalk in the prior knowledge network, but also to the biological importance of signaling that is not confined to what was more or less arbitrarily viewed as pathways.

Model topology parameterization

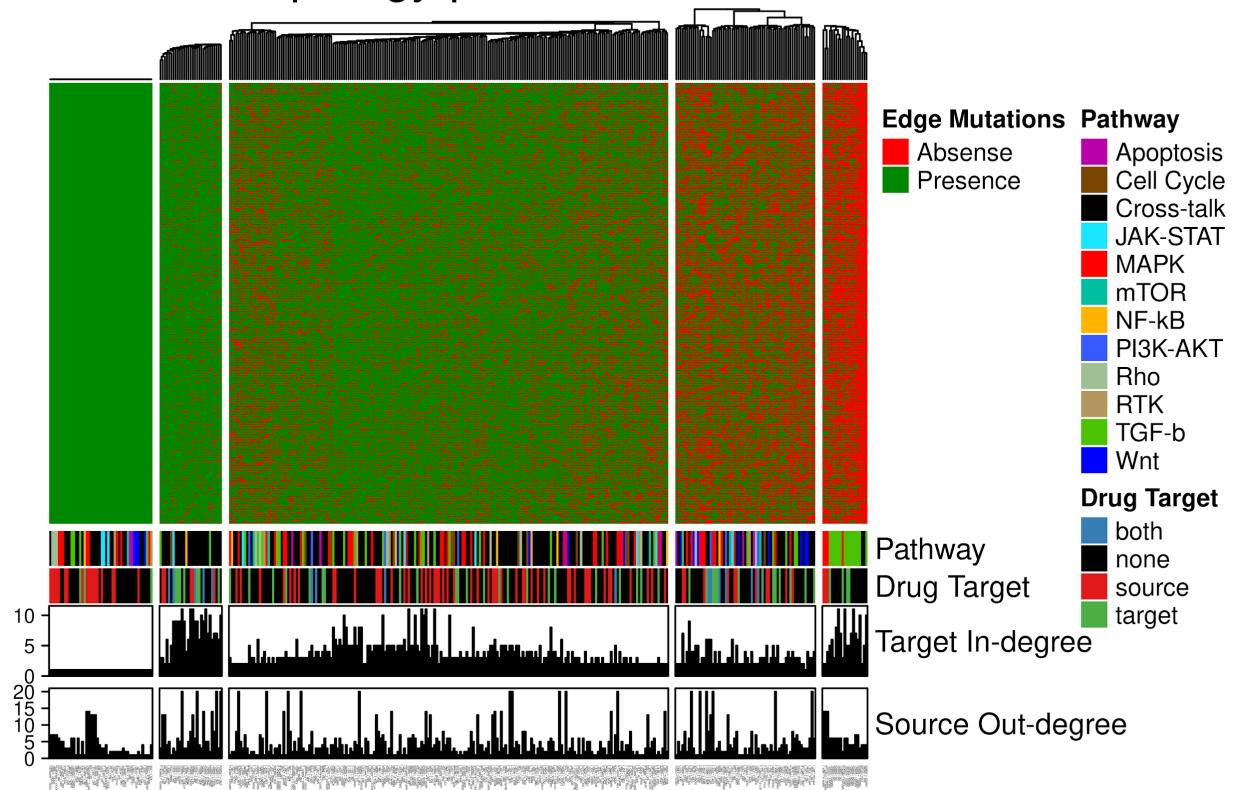


Figure 10: Heatmap of steady state models with topology mutations. Each column represents an interaction, each row represents one model, all rows jointly represent the ensemble. Five clusters, guided by K-means clustering, stand out: the first cluster from the left represents edges that cannot be removed because nodes would lose regulations. The second cluster from left represents nodes that are likely to be preserved, the middle cluster represents edges that are often preserved, the fourth cluster from left represents edges that are often discarded, and the last cluster, right, represents edges that are almost always discarded in the evolution to maximum fitness.

Validation of synergy predictions in vivo

In order to test the translational relevance of our drug synergy prediction platform we performed in vivo validation for one of the proposed novel drug synergies. The synergies of TAK1 inhibition combined with PI3K inhibition, already identified in our previous logical modeling work, had not been reported earlier and thus represent novel synergies of potential interest in future cancer therapy. We rediscover the same synergy of combined TAK1 and PI3K inhibition in our framework for automated model parameterization.

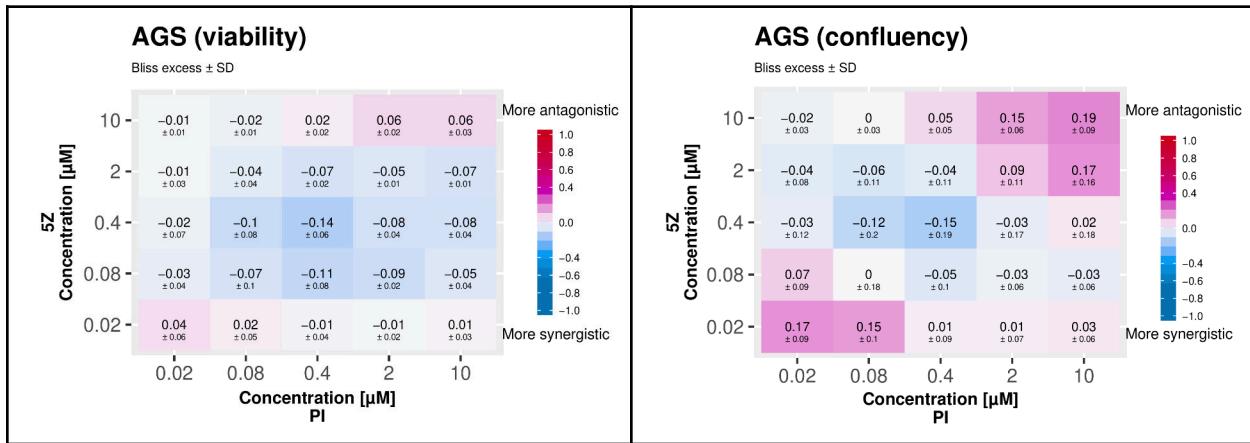


Figure 11: Synergy of the drugs targeting TAK1 (5Z) and PI3K (PI) confirmed in viability (left) and confluency (right) screens.

In order to test reproducibility across different high throughput drug screen readouts we subjected AGS cancer cells to combined TAK1 and PI3K inhibition and monitored the response both by ATP content measurement (viability) and by microscopy (confluence), see Figure 11. For both readouts there is a region of synergistic response to drugs applied at medium doses as indicated by the drug concentration gradients. We subcutaneously injected AGS gastric adenocarcinoma xenograft tumors in Balb/c mice to test the synergy of combined inhibition of TAK1 (5Z-7-oxozeaenol) and PI3K (PI103) *in vivo*. The xenograft tumors ($n=30$) were randomized to four groups: control, PI103, (5Z)-7-oxozeaenol and a combination group which received both PI103 and (5Z)-7-oxozeaenol, see figure 12A. At the end of the experiment the combination group displayed significant changes (t-test) in relative tumor size compared to either single-treatment group (Figure 12B and 12C). We observed a similar reduced proliferative capacity for tumors in mice treated jointly with TAK1 and PI3K inhibitors, as indicated by tumour proliferation marker Ki67 (Figure 12D and 12E). The clear difference between the significant tumor growth inhibitory effect of the combination and the non-significant activity of individual agents strongly indicates a synergistic anti-tumor effect of the two agents. A concern for drug synergies is that possible side-effects might also be expressed in synergistic ways. We therefore chose doses to be at the lower end of effective concentrations, compared to previously

published in vivo use of the inhibitors^{33, 34, 35, 36}. Despite low dosage the inhibitors together reduced tumor growth, without signs of pain or weight loss.

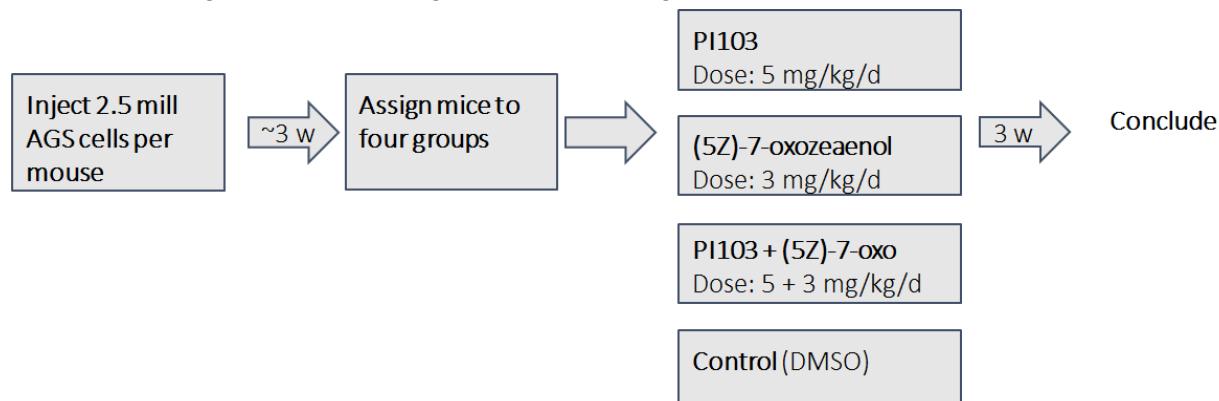


Figure 12A: Xenograft experimental design. Cancer cells were injected subcutaneously in mice and allowed to grow until a visible tumour could be identified, after which the mice were randomized into four groups. Mice were then treated with single drugs and the combination for 19 days after which the experiment was stopped and tumor sizes evaluated.

³³ Bhattacharya, B., Akram, M., Balasubramanian, I., Tam, K. K. Y., Koh, K. X., Yee, M. Q., & Soong, R. (2012). Pharmacologic synergy between dual phosphoinositide-3-kinase and mammalian target of rapamycin inhibition and 5-fluorouracil in PIK3CA mutant gastric cancer cells. *Cancer Biology & Therapy*, 13(1), 34–42. <https://doi.org/10.4161/cbt.13.1.18437>

³⁴ Donev, I. S., Wang, W., Yamada, T., Li, Q., Takeuchi, S., Matsumoto, K., ... Yano, S. (2011). Transient PI3K inhibition induces apoptosis and overcomes HGF-mediated resistance to EGFR-TKIs in EGFR mutant lung cancer. *Clinical Cancer Research : An Official Journal of the American Association for Cancer Research*, 17(8), 2260–2269. <https://doi.org/10.1158/1078-0432.CCR-10-1993>

³⁵ Fan, Y., Cheng, J., Vasudevan, S. a, Patel, R. H., Liang, L., Xu, X., ... Yang, J. (2013). TAK1 inhibitor 5Z-7-oxozeanol sensitizes neuroblastoma to chemotherapy. *Apoptosis : An International Journal on Programmed Cell Death*, 18(10), 1224–1234. <https://doi.org/10.1007/s10495-013-0864-0>

³⁶ Singh, A., Sweeney, M. F., Yu, M., Burger, A., Greninger, P., Benes, C., ... Settleman, J. (2012). TAK1 inhibition promotes apoptosis in KRAS-dependent colon cancers. *Cell*, 148(4), 639–650. <https://doi.org/10.1016/j.cell.2011.12.033>

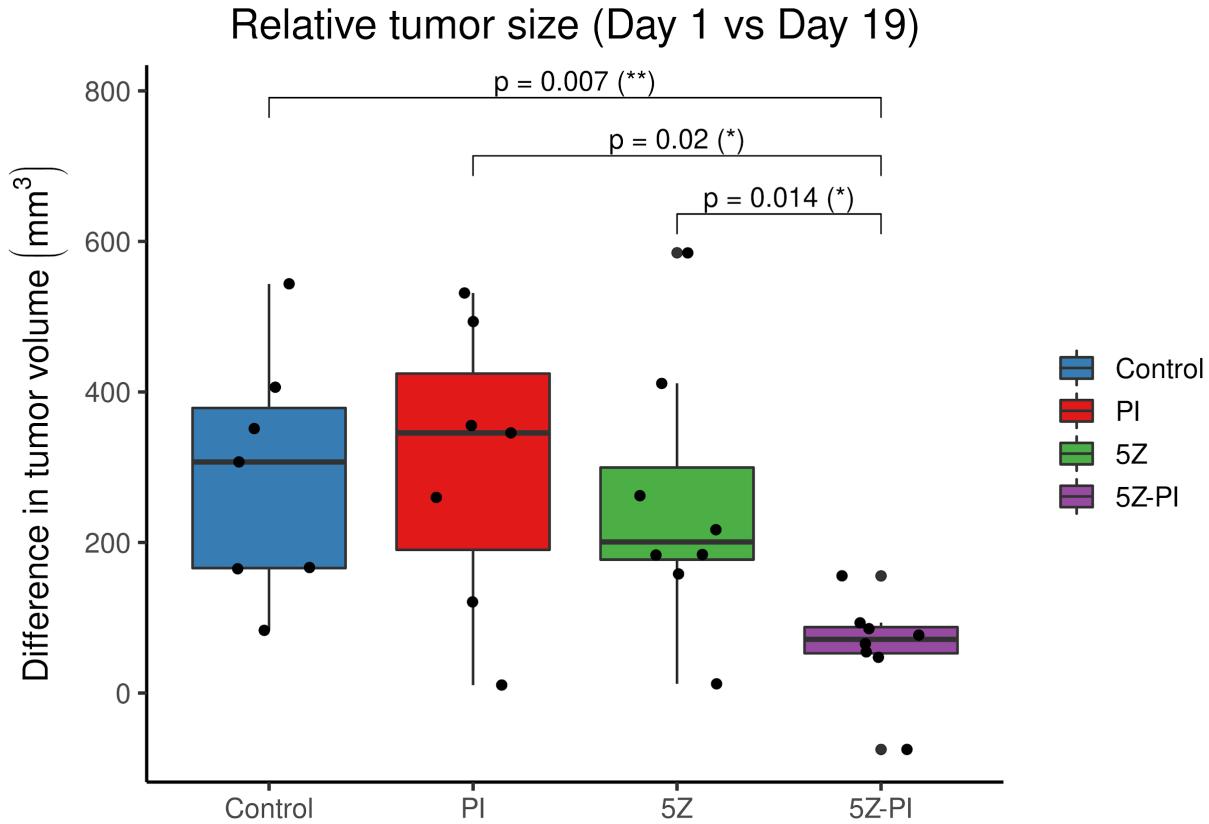


Figure 12B: Testing of drug synergy in a mouse xenograft model. Tumor volumes determined at the end of the study were compared with tumor volumes at treatment onset. Tumors in mice receiving both inhibitors 5Z-7-oxozeaenol 3 mg/kg/d (5Z) and PI103 5 mg/kg/d (PI) show a smaller increase in size compared to either of the groups receiving only single inhibitors, and the control group. The combination effect was statistically significantly different from either single drug therapy as evaluated by Mann-Whitney U tests with corresponding p-values shown above the boxplots. Similar results were obtained using t-tests with a chosen significance level of $p=0.05$.

Average tumor volume with SEM

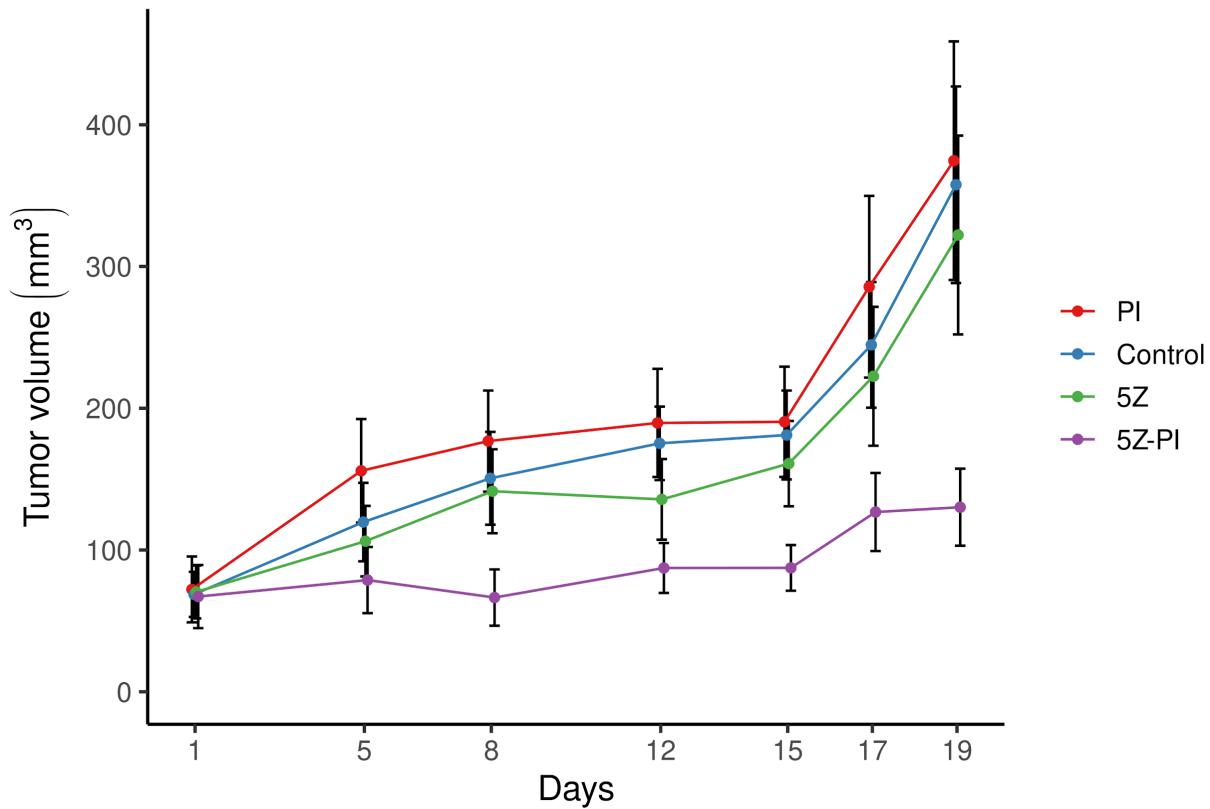


Figure 12C: Average tumor volume for the four groups of mice with standard error of the mean (SEM) indicated by the error bars. The group receiving both inhibitors (5Z + PI) displays a more inhibited tumor growth than either of the groups receiving each single inhibitor and the control group.

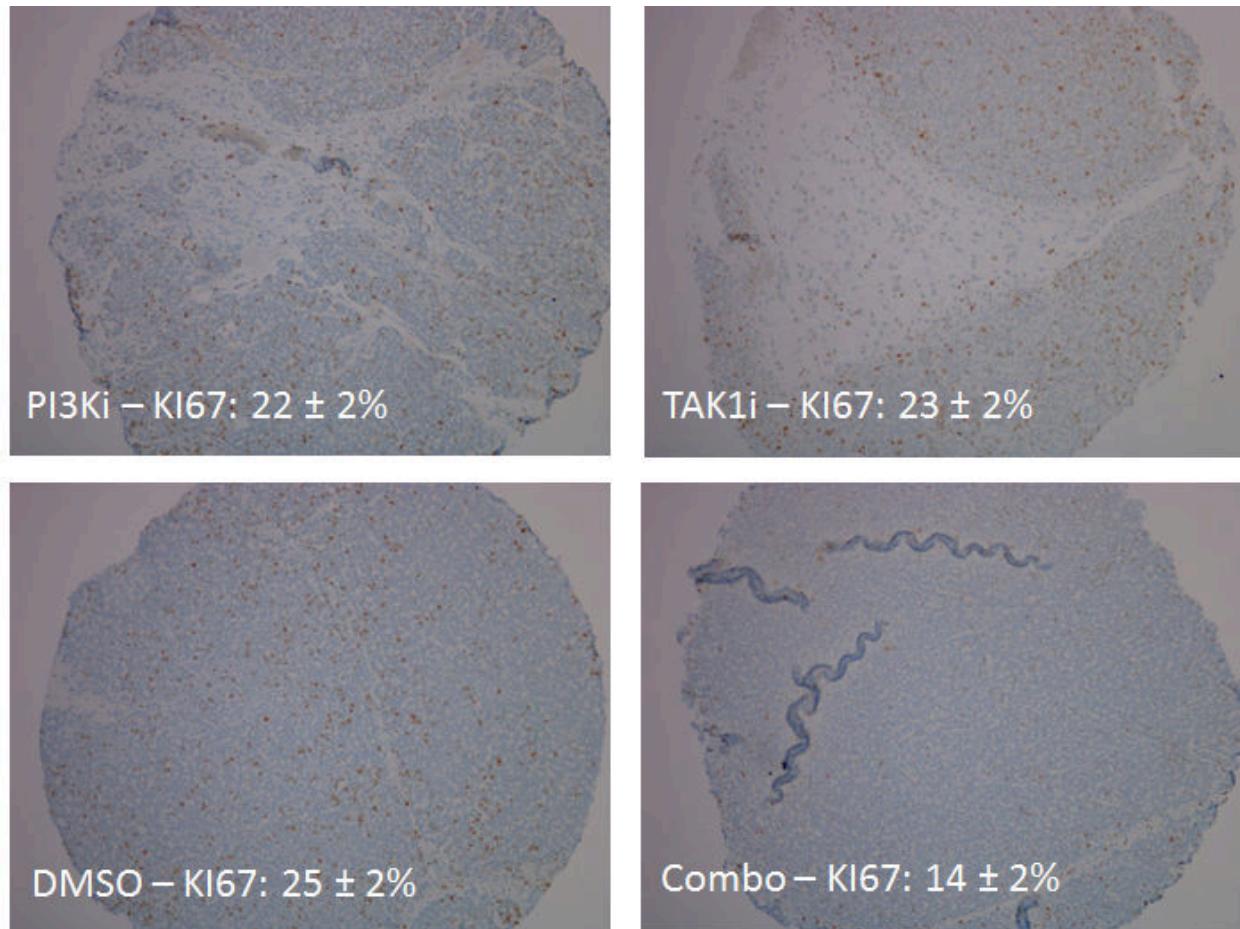


Figure 12D: Ki67 proliferation index (count of Ki67 positive cells) reduced upon joint combination of TAK1 and PI3K inhibition.

Discussion

Our results show that a curated prior knowledge network with an initial set of logical rules can be automatically parameterized by a genetic algorithm, using a fitness score reflecting how well the global stable state of a model matches the experimentally determined local states of signaling components of a cell line in its native growing state. We showed that the predictive performance from an automatically parameterized ensemble of models was on par with our original, curated CASCADE 1.0 model. We next used our parameterization software to calibrate a larger network topology. With larger topologies, benefits of automation become more apparent, and can be used to enable simulation for larger numbers of drugs and numbers of cell lines.

Finding drug synergies among the vast set of possible combinations of drugs calls for new approaches. To rationally reduce the prohibitively large experimental search space, we found that our approach can be highly useful to identify sets of drugs that are unlikely to display synergy and that should not be prioritized for testing. Even tackling the combinatorial complexity for standardized cancer models is already challenging, as exemplified by the AZS-DREAM Challenge³⁷, where pairwise combinations of 118 drugs (6903 possible drug-drug combinations) are tested against a panel of 85 cell lines. If all combinations are screened in 6x6 matrices this corresponds to over 200.000 384-well plates for four technical replicates, clearly indicating that a trial-and-error approach is not economic for drug synergy discovery. Whereas most approaches for drug synergy predictions rely on perturbation data for training a classifier^{38, 39, 40, 41}, our approach works well with calibration data based only on data from an unperturbed system, which greatly reduces the cost of data acquisition and opens possibilities for clinical applications.

Assessing the performance of drug synergy predictions is met with several challenges. First, targeted drugs, as those employed here ('small molecule' inhibitors), can affect a number of other targets in addition to their intended target, known as 'off-target' effects⁴². Since our simulation is based on canonical drug targets annotated for each drug, any information missing about off-target effects must be expected to impact simulations. In addition, drug synergy is an elusive concept itself, with different mathematical reference models producing different synergy scores⁴³. Finally, high throughput drug screens, as employed here, typically reports drug responses based on measured residual ATP content after drug exposure, which is known not to

³⁷ Menden, M. P., Wang, D., Guan, Y., Mason, M., Bulusu, K. C., Yu, T., ... Saez-rodriguez, J. (2017). Community assessment of cancer drug combination screens identifies strategies for synergy prediction. <http://doi.org/10.1101/200451>

³⁸ Preuer, K., Lewis, R. P. I., Hochreiter, S., Bender, A., Bulusu, K. C., & Klambauer, G. (2017). DeepSynergy: Predicting anti-cancer drug synergy with Deep Learning. Bioinformatics (Oxford, England), (December), 1–9. <http://doi.org/10.1093/bioinformatics/btx806>

³⁹ Yang, J., Tang, H., Li, Y., Zhong, R., Wang, T., Wong, S. T. C., ... Xie, Y. (2015). DIGRE: Drug-induced genomic residual effect model for successful prediction of multidrug effects. CPT: Pharmacometrics and Systems Pharmacology, 4(2), 91–97. <http://doi.org/10.1002/psp4.1>

⁴⁰ Ianevski, A., Giri, A. K., Gautam, P., Kononov, A., Potdar, S., Saarela, J., ... Aittokallio, T. (2019). Prediction of drug combination effects with a minimal set of experiments. Nature Machine Intelligence, 1(12), 568–577. <http://doi.org/10.1038/s42256-019-0122-4>

⁴¹ Fröhlich, F., Kessler, T., Weindl, D., Shadrin, A., Schmiester, L., Hache, H., ... Hasenauer, J. (2018). Efficient Parameter Estimation Enables the Prediction of Drug Response Using a Mechanistic Pan-Cancer Pathway Model. Cell Systems, 7, 1–13. <http://doi.org/10.1016/j.cels.2018.10.013>

⁴² Klaeger, S., Heinzelmeir, S., Wilhelm, M., Polzer, H., Vick, B., Koenig, P.-A., ... Kuster, B. (2017). The target landscape of clinical kinase drugs. Science (New York, N.Y.), 358(6367), eaan4368. <https://doi.org/10.1126/science.aan4368>

⁴³ Vlot, A. H. C., Aniceto, N., Menden, M. P., Ulrich-Merzenich, G., & Bender, A. (2019). Applying drug synergy metrics to oncology combination screening data: agreements, disagreements and pitfalls. Drug Discovery Today, 00(00). <https://doi.org/10.1016/j.drudis.2019.09.002>

capture all growth-reducing drug responses^{44, 45, 46}. Despite these limitations, which must be expected to reduce the performance of any drug synergy prediction approach, our logical simulation-based in-silico pre-selection approach performs immensely better than a blinded screen that would assay the same numbers of candidates: at a sensitivity of 50%, roughly 35-40% of a pre-selected set of predicted synergies will be observed in follow up drug synergy experiments in drug screen where only 4% of drug combinations acted synergistically overall.

Our choice of a logical framework for computational simulations comes both with some benefits and limitations. Logic equations are very quick to evaluate, with high simulation speed enabling extensive simulations even on regular desktop computers. However, logic equations as employed here only allow two activity states for model components: active and inactive. Moreover, only two interaction strengths between components are allowed: full interaction or no interaction. These limitations, however, still to a large extent meet the demands and possibilities offered by experiments with present day laboratory techniques. Data from these experiments often lack finer-grained observations that would be needed for continuous modeling approaches and therefore logical modeling represents a valid compromise between molecular data richness and computation speed. Our implementation of logical model simulations only computes stable states, thereby discarding any potential complex attractor of models. This choice was based on computational efficiency, as computation of complex attractor in addition to stable states would severely tax our simulations. One possible avenue for future research will be to account for also complex attractors, either by approximations as offered by e.g. trap space analysis, or by a full characterization of model behaviour.

We find that our approach is somewhat sensitive to errors in the calibration data, and even more sensitive to errors in the prior knowledge, indicating that curation quality is paramount to our modeling approach. This demands for adequate causal statement curation protocols and

⁴⁴ Bae, S. Y., Guan, N., Yan, R., Warner, K., Taylor, S. D., & Meyer, A. S. (2020). Measurement and models accounting for cell death capture hidden variation in compound response. *Cell Death and Disease*, 11(4). <https://doi.org/10.1038/s41419-020-2462-8>

⁴⁵ Gautam, P., Karhinan, L., Szwajda, A., Jha, S. K., Yadav, B., Aittokallio, T., & Wennerberg, K. (2016). Identification of selective cytotoxic and synthetic lethal drug responses in triple negative breast cancer cells. *Molecular Cancer*, 15(1), 1–16. <https://doi.org/10.1186/s12943-016-0517-3>

⁴⁶ Folkesson, E., Niederdorfer, B., Nakstad, V. T., Thommesen, L., Klinkenberg, G., Lægreid, A., & Flobak, Å. (2020). High-throughput screening reveals higher synergistic effect of MEK inhibitor combinations in colon cancer spheroids. *Scientific Reports*, 10(1), 11574. <https://doi.org/10.1038/s41598-020-68441-0>

standards that feed into high quality general and cancer signaling databases, a demand that is materializing^{47, 48, 49, 50, 51}.

Interestingly, we found that deleting a small fraction of the regulatory links from the prior knowledge network can be very powerful in optimizing models for drug synergy predictions, whereas randomly rearranging regulatory links is very detrimental to model performance. Logical model construction can be performed by curating data resources or the literature, or by relying on high quality curated databases like Signor, SignalLink or IntAct. If quality of these resources, or the ad hoc curation of literature, would not be of high standard, this would significantly limit the performance of the resulting model. On the other hand, calling a prior knowledge network complete is essentially a judgement call, as all models are limited. This demonstrates that, while lacking in completeness, high quality curation can produce models that can predict drug synergies. Furthermore, whereas the inclusion of a regulatory link ideally needs evidence from observations about the functional relevance of such a link in the cell that is modeled, our observations about ERK activity highlight the variability of experimental data concerning such observations. It is therefore not unreasonable to accept that an optimization algorithm can choose to dismiss regulatory links for the benefit of improved model performance. The increasing availability of high quality curated molecular causal interaction data opens a perspective to fully automated model building, where algorithmic topology optimization can fine tune a model to perform adequately, for any target node requirements and cell type for which baseline biomarker data is available. A welcome feature of automatically parameterized logical models is their innate ability to suggest mechanisms underpinning a particular observation. Such model-driven hypotheses can lay foundations for targeted follow-up experiments that provide observations for directed model revisions, resulting in a model with higher validity.

⁴⁷ Licata, L., Lo Surdo, P., Iannuccelli, M., Palma, A., Micarelli, E., Perfetto, L., ... Cesareni, G. (2020). SIGNOR 2.0, the SIGnaling Network Open Resource 2.0: 2019 update. *Nucleic Acids Research*, 48(D1), D504–D510. <http://doi.org/10.1093/nar/gkz949>

⁴⁸ Türei, D., Korcsmáros, T., & Saez-Rodriguez, J. (2016). OmniPath: guidelines and gateway for literature-curated signaling pathway resources. *Nature Methods*, 13(12), 966–967. <http://doi.org/10.1038/nmeth.4077>

⁴⁹ Touré, V., Vercruyse, S., Acencio, M. L., Lovering, R. C., Orchard, S., Bradley, G., ... Kuiper, M. (2020). The Minimum Information about a Molecular Interaction Causal Statement (MI2CAST). *Bioinformatics* (Oxford, England), 1–8. <https://doi.org/10.1093/bioinformatics/btaa622>

⁵⁰ Touré, V., Zobolas, J., Kuiper, M., & Vercruyse, S. (2021). CausalBuilder: bringing the MI2CAST causal interaction annotation standard to the curator. *Database : The Journal of Biological Databases and Curation*, 2021, 1–6. <https://doi.org/10.1093/database/baaa107>

⁵¹ Cristobal Monraz Gomez, L., Kondratova, M., Ravel, J. M., Barillot, E., Zinovyev, A., & Kuperstein, I. (2019). Application of Atlas of Cancer Signalling Network in preclinical studies. *Briefings in Bioinformatics*, 20(2), 701–716. <https://doi.org/10.1093/bib/bby031>

It has been suggested that network topology alone already explains drug synergy^{52,53}, and that parameterization to a lesser extent defines or refines synergy predictions⁵⁴. Experimentally it has been observed that drug synergies tend to vary between cell lines, with the most frequently observed synergistic drug pair only effective in about half of the cell lines analyzed^{55,56}. In our analysis we observed that drug synergy predictions depend both on the specific parameterization of a given topology and on the topology itself. When we put our manually defined topology to the test, drug synergy predictions are more accurate for models optimized to represent cell-specific baseline biomarkers in their local states, compared to unconstrained local states. One may speculate that the interactions relevant to describe drug combination effects represent a subset of all potential (general) interactions, and that this subset varies from cell line to cell line. From a completeness perspective, given the limited knowledge of molecular biology today, any model representation will be a major simplification of reality, yet some of these models work.

⁵² Cokol, M., Chua, H. N., Tasan, M., Mutlu, B., Weinstein, Z. B., Suzuki, Y., ... Roth, F. P. (2011). Systematic exploration of synergistic drug pairs. *Molecular Systems Biology*, 7(544), 544. <http://doi.org/10.1038/msb.2011.71>

⁵³ Jaeger, S., Igea, A., Arroyo, R., Alcalde, V., Canovas, B., Orozco, M., ... Aloy, P. (2017). Quantification of pathway cross-talk reveals novel synergistic drug combinations for breast cancer. *Cancer Research*, 77(2), 459–469. <http://doi.org/10.1158/0008-5472.CAN-16-0097>

⁵⁴ Yin, N., Ma, W., Pei, J., Ouyang, Q., Tang, C., & Lai, L. (2014). Synergistic and Antagonistic Drug Combinations Depend on Network Topology. *PLoS ONE*, 9(4), e93960. <http://doi.org/10.1371/journal.pone.0093960>

⁵⁵ Axelrod, M., Gordon, V. L., Conaway, M., Tarcsafalvi, A., Neitzke, D. J., Gioeli, D., & Weber, M. J. (2013). Combinatorial drug screening identifies compensatory pathway interactions and adaptive resistance mechanisms. *Oncotarget*, 4(4), 622–35. <http://doi.org/10.18632/oncotarget.938>

⁵⁶ <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2642-7>

Supplementary Figures

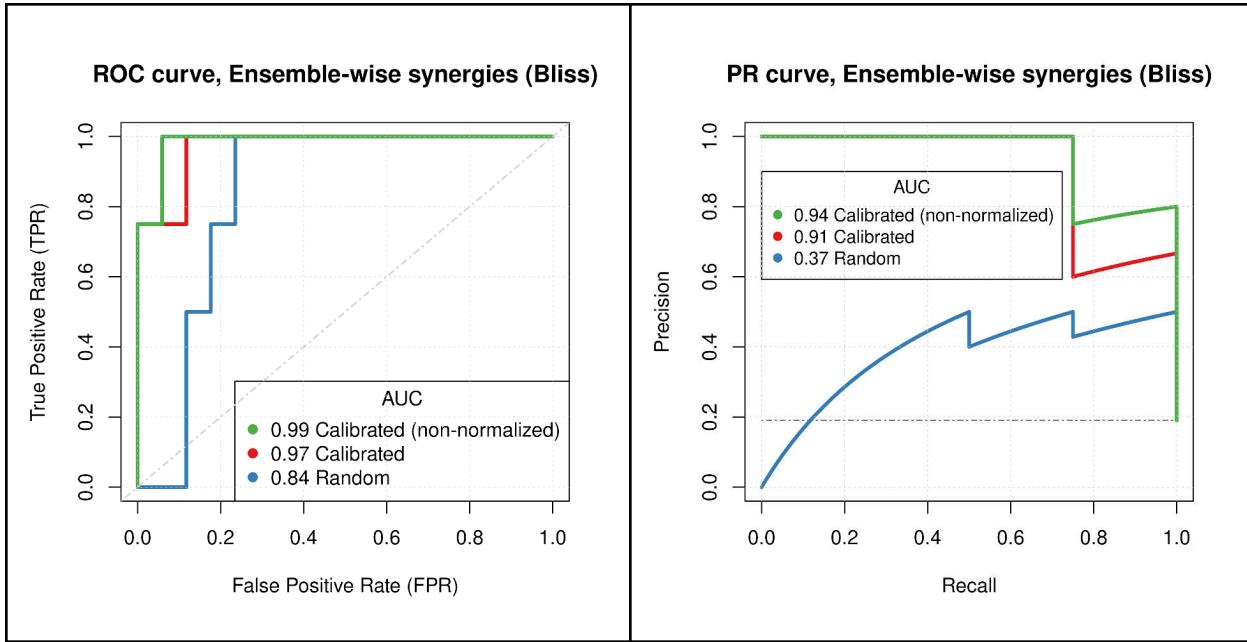


Figure S1: Predictive performance for random (proliferative) models, calibrated models (non-normalized) and calibrated normalized to random models (CASCADE 1.0 topology).

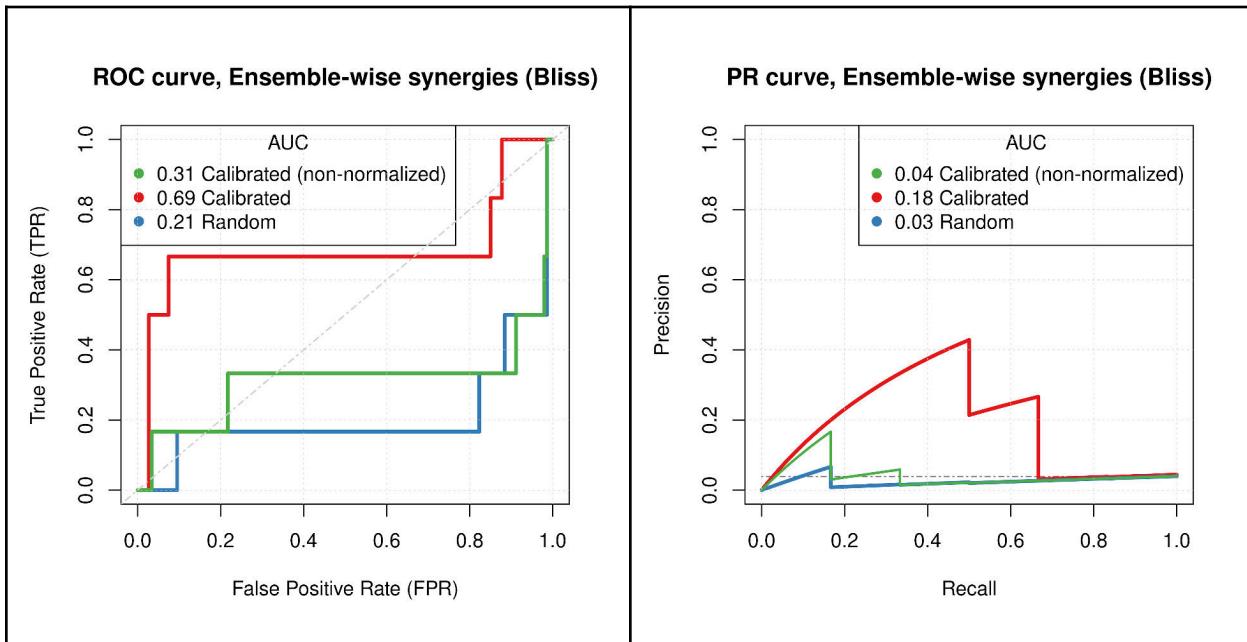


Figure S2: Predictive performance for random (proliferative) models, calibrated models (non-normalized) and calibrated normalized to random models (CASCADE 2.0 topology).

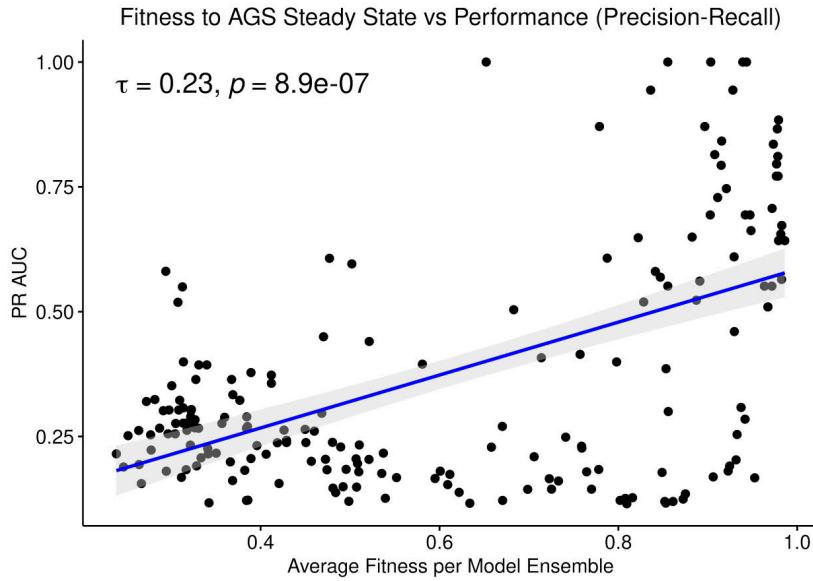


Figure S3: PR AUC performance dependence on fitness (CASCADE 1.0 topology).

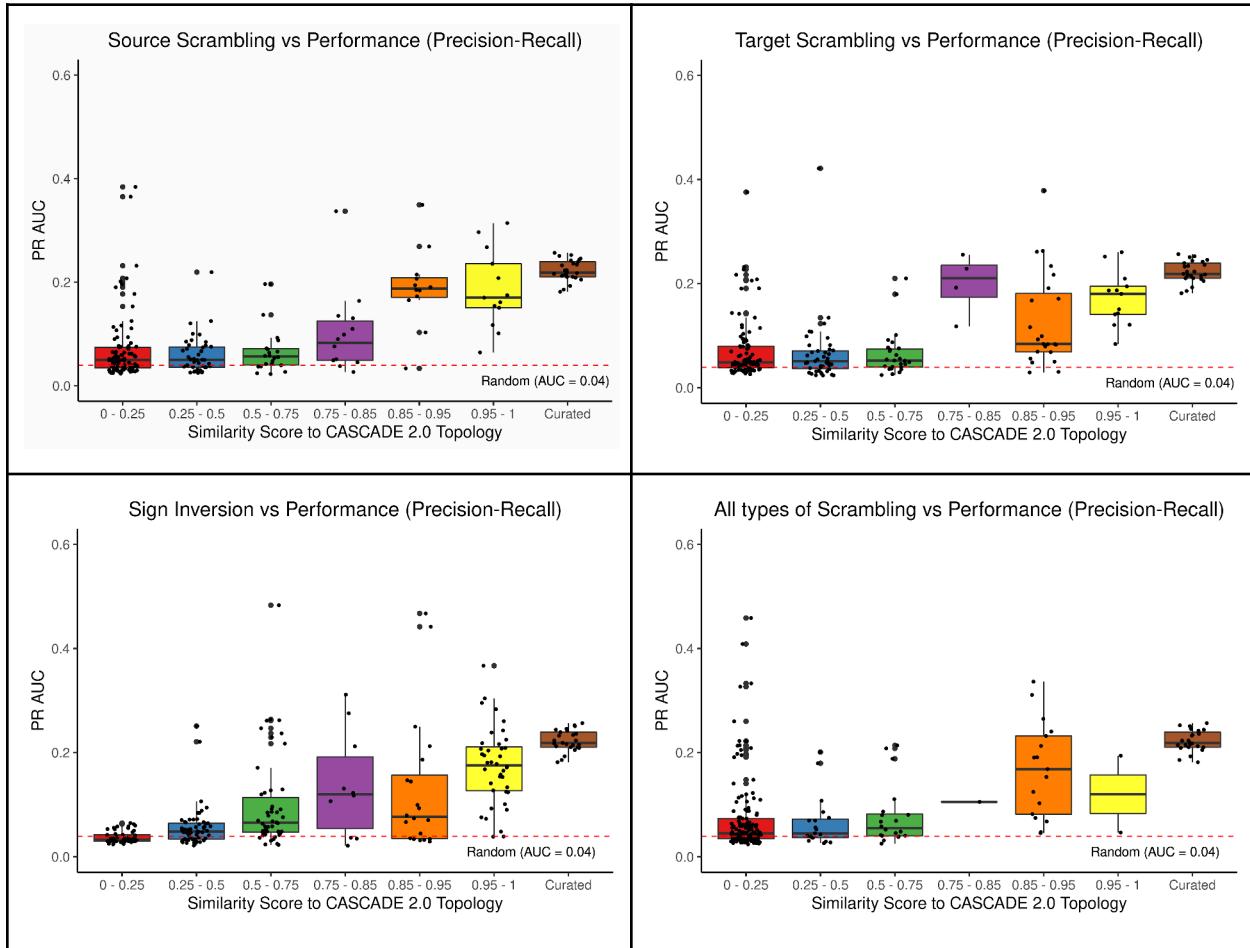


Figure S4: Effects of variations introduced in the CASCADE 2.0 prior knowledge graph (PR AUC performance metric).

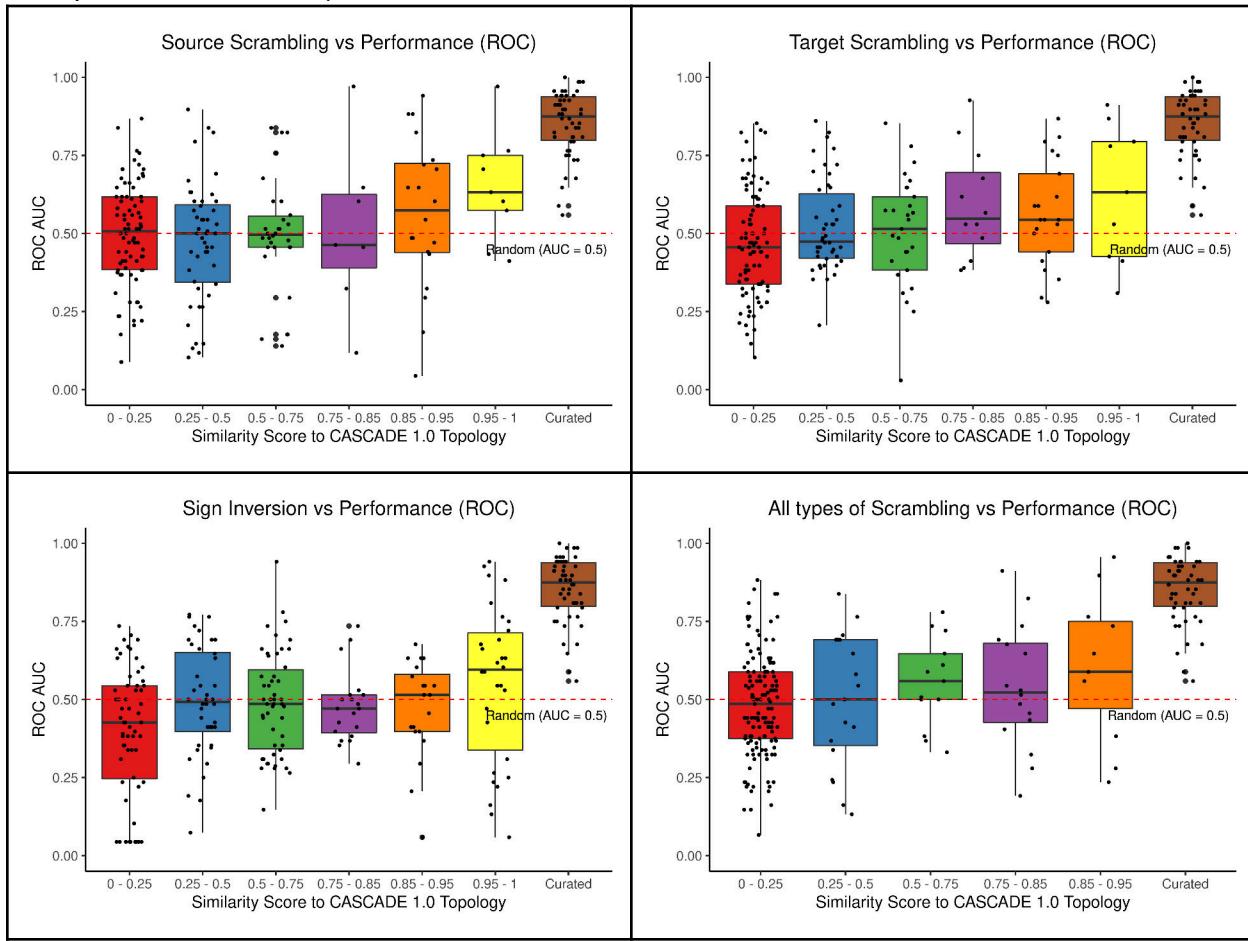


Figure S5: Effects of variations introduced in the CASCADE 1.0 prior knowledge graph (ROC AUC performance metric).

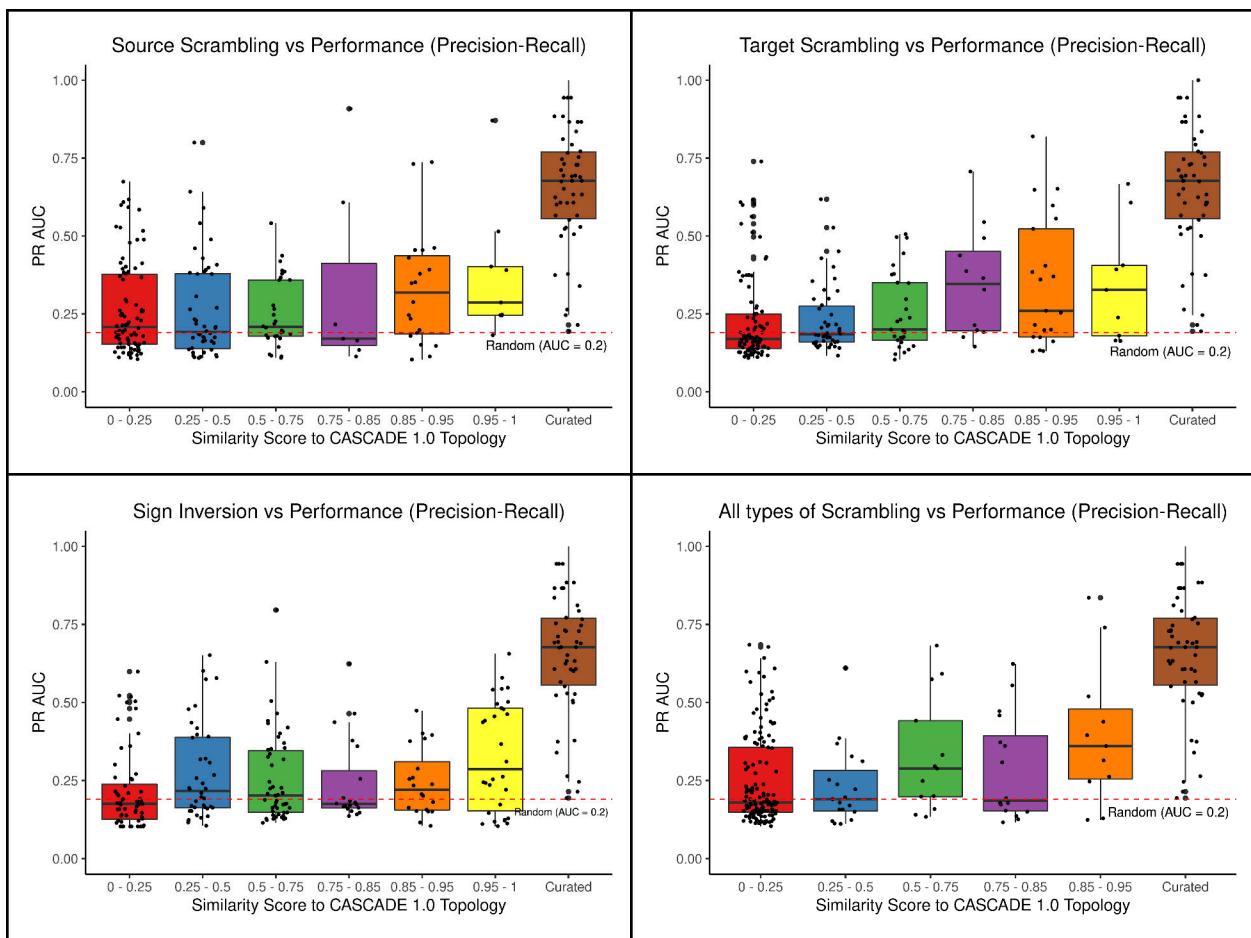


Figure S6: Effects of variations introduced in the CASCADE 1.0 prior knowledge graph (PR AUC performance metric).

PAPER 4

emba: R package for analysis and visualization of biomarkers in boolean model ensembles

John Zobolas^{1, 2}, Martin Kuiper¹, and Åsmund Flobak^{2, 3}

¹ Department of Biology, Norwegian University of Science and Technology (NTNU), Trondheim, Norway ² Department of Clinical and Molecular Medicine, Norwegian University of Science and Technology (NTNU), Trondheim, Norway ³ The Cancer Clinic, St. Olav's Hospital, Trondheim, Norway

DOI: [10.21105/joss.02583](https://doi.org/10.21105/joss.02583)

Software

- [Review ↗](#)
- [Repository ↗](#)
- [Archive ↗](#)

Editor: Mikkel Meyer Andersen
✉

Reviewers:

- [@neerajdhanraj](#)
- [@edifice1989](#)

Submitted: 28 July 2020

Published: 26 September 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Introduction

Computational modeling of cellular systems has been one of the most powerful tools used to build interpretable knowledge of biological processes and help identify molecular mechanisms that drive diseases such as cancer (Aldridge, Burke, Lauffenburger, & Sorger, 2006). In particular, the use of logical modeling has proven to be a substantially useful approach, since it allows the easy construction, simulation and analysis of predictive models, capable of providing a qualitative and insightful view on the extremely complex landscape of biological systems (Abou-Jaoudé et al., 2016; Morris, Saez-Rodriguez, Sorger, & Lauffenburger, 2010; Wang, Saadatpour, & Albert, 2012). Such mechanistic models, with the systematic integration of prior knowledge and experimental data, have been extensively used to better understand what drives deregulation of signal transduction, the outcome of which is the manifestation of diseases (Traynard, Tobalina, Eduati, Calzone, & Saez-Rodriguez, 2017). Furthermore, their explanatory power has been used to provide insights into a drug's mode of action, investigate the mechanisms of resistance to drugs (Eduati et al., 2017) and suggest new therapeutic combination candidates, among others (Flobak et al., 2015).

One of the major challenges in systems medicine, has been the identification of scientifically validated, predictive biomarkers that correlate with patient response to given therapies. The analysis of biological predictive markers of pharmacologic response can not only further our understanding of the systemic processes involved in diseases but can also help to classify patients into groups with similar responses to specific therapeutic interventions, advancing personalized medicine (Senft, Leiserson, Ruppin, & Ronai, 2017). In addition, the identification of biomarkers in tumor cells (e.g. mutations) has enabled the discovery of drug targets which are utilized in combinatorial molecular-targeted therapies - a strategy which aims to treat specific patient subgroups and has shown larger overall survival rates and reduced side-effects than monotherapy (Al-Lazikani, Banerji, & Workman, 2012). Despite the huge advancements towards drug combination therapy, genetic heterogeneity, drug resistance and drug combination synergy mechanisms still pose fundamental challenges to clinicians, modelers and lab researchers.

To help bridge the model simulation results with the (clinical) laboratory observations, several optimization methods have been used, such as model calibration, parameter estimation and sensitivity analysis. These methods also allow us to determine which model parameters have the biggest influence in the overall behaviour of the system (Aldridge et al., 2006). For example, in Fröhlich et al. (2018), a computational framework that allowed for the efficient parameterization and contextualization of a large-scale cancer signaling network, was used to predict combination treatment outcome from single drug data. This model was calibrated to fit and accurately describe specific cell-line experimental data, while enabling the identification

of biomarkers of drug sensitivity as well as molecular mechanisms that affect drug resistance. Furthermore, in Dorier et al. (2016), a network optimization approach which topologically parameterized boolean models according to a genetic algorithm was used, in order to best match the experimentally observed behaviour. This method resulted in an ensemble of boolean models which can be used to simulate response under drug perturbations in order to assess the underlying mechanisms and to generate new testable hypotheses. Such an aggregation of best-fit models (wisdom of the crowds) has been shown to be quite robust and effective for model prediction performance (Marbach et al., 2012).

Statement of need

There is a plethora of software tools devoted to the qualitative modeling and analysis of biological networks. The Consortium for the development of Logical Models and Tools (CoLoMoTo) is a community effort which aims to standardize the representation of logical networks and provide a common repository of methods and tools to analyze these networks (Naldi et al., 2015). Furthermore, to facilitate the access to several software logical modeling tools and enable reproducible computational workflows, the CoLoMoTo Interactive Notebook was introduced as a unified computational framework (Aurélien Naldi, Hernandez, Levy, et al., 2018). The incorporated tools are accessed via a common programming interface (though originally implemented in different programming languages e.g. Java, Python, C++ and R) and offer a collection of features like accessing online model repositories (Helikar et al., 2012), model editing (Aurélien Naldi, Hernandez, Abou-Jaoudé, et al., 2018), dynamical analysis (finding attractors, stochastic simulations, reachability properties, model-checking techniques) (Klarner, Streck, Siebert, & Sahinalp, 2016; Müssel, Hopfensitz, & Kestler, 2010; Aurélien Naldi, 2018; Paulevé, 2017; Stoll et al., 2017) and model parameterization/optimization to fit perturbation signaling data (Gjerga et al., 2020; Terfve et al., 2012). Despite the diverse and multi-purpose logical modeling tools that exist, there is still a lack of data analysis-oriented software that assists with the discovery of predictive biomarkers in ensembles of parameterized boolean networks that have been subject to drug combination perturbations.

The `emba` R package aims to fill that gap and provide a first implementation of such a novel software. Initially, it was designed as a complementary software tool, to help the analysis of the parameterized boolean model ensembles which were produced by modules from the DrugLogics NTNU software pipeline (see respective documentation (Zobolas, 2020a)). Later, we generalized most of the functions in the package and modularized them to package-essential (that form the core of the `emba` package) and various general-purpose yet useful functions (that are now part of the dependency package `usefun` (Zobolas, 2020b)).

Summary

The main functionality of the `emba` R package is to find *performance* and *synergy* biomarkers. Performance biomarkers are nodes in the input boolean networks whose activity state and/or model parameterization affects the predictive performance of those models. The prediction performance can be assessed via the number of true positive predictions or the Matthews correlation coefficient score which is more robust to class imbalances (Chicco & Jurman, 2020). On the other hand, synergy biomarkers are nodes which provide hints for the mechanisms behind the complex process of synergy manifestation in drug combination datasets.

For more information, see our “Get started guide” and the reference manual in the package website (Zobolas, 2020c). Several analyses using the `emba` R package are available in a separate repository (Zobolas, 2020d). Future developments will include the implementation of a method for the identification of *topology* biomarkers, where we will be able to assess

which interactions in the network are important for the manifestation of synergies in specific cell-contexts.

Acknowledgements

This work was supported by ERACoSysMed grant *COLOSYS* (JZ, MK) and The NTNU Strategic Research Area *NTNU Health* (AF).

References

- Abou-Jaoudé, W., Traynard, P., Monteiro, P. T., Saez-Rodriguez, J., Helikar, T., Thieffry, D., & Chaouiya, C. (2016). Logical Modeling and Dynamical Analysis of Cellular Networks. *Frontiers in genetics*, 7, 94. doi:[10.3389/fgene.2016.00094](https://doi.org/10.3389/fgene.2016.00094)
- Aldridge, B. B., Burke, J. M., Lauffenburger, D. A., & Sorger, P. K. (2006). Physicochemical modelling of cell signalling pathways. *Nature Cell Biology*, 8(11), 1195–1203. doi:[10.1038/ncb1497](https://doi.org/10.1038/ncb1497)
- Al-Lazikani, B., Banerji, U., & Workman, P. (2012). Combinatorial drug therapy for cancer in the post-genomic era. *Nature Biotechnology*, 30(7), 679–692. doi:[10.1038/nbt.2284](https://doi.org/10.1038/nbt.2284)
- Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1). doi:[10.1186/s12864-019-6413-7](https://doi.org/10.1186/s12864-019-6413-7)
- Dorier, J., Crespo, I., Niknejad, A., Liechti, R., Ebeling, M., & Xenarios, I. (2016). Boolean regulatory network reconstruction using literature based knowledge with a genetic algorithm optimization method. *BMC Bioinformatics*, 17(1). doi:[10.1186/s12859-016-1287-z](https://doi.org/10.1186/s12859-016-1287-z)
- Eduati, F., Doldà-Martelli, V., Klinger, B., Cokelaer, T., Sieber, A., Kogera, F., Dorel, M., et al. (2017). Drug Resistance Mechanisms in Colorectal Cancer Dissected with Cell Type-Specific Dynamic Logic Models. *Cancer research*, 77(12), 3364–3375. doi:[10.1158/0008-5472.CAN-17-0078](https://doi.org/10.1158/0008-5472.CAN-17-0078)
- Flobak, Å., Baudot, A., Remy, E., Thommesen, L., Thieffry, D., Kuiper, M., & Lægreid, A. (2015). Discovery of Drug Synergies in Gastric Cancer Cells Predicted by Logical Modeling. (I. Xenarios, Ed.) *PLOS Computational Biology*, 11(8), e1004426. doi:[10.1371/journal.pcbi.1004426](https://doi.org/10.1371/journal.pcbi.1004426)
- Fröhlich, F., Kessler, T., Weindl, D., Shadrin, A., Schmiester, L., Hache, H., Muradyan, A., et al. (2018). Efficient Parameter Estimation Enables the Prediction of Drug Response Using a Mechanistic Pan-Cancer Pathway Model. *Cell Systems*, 7(6), 567–579.e6. doi:[10.1016/J.CELS.2018.10.013](https://doi.org/10.1016/J.CELS.2018.10.013)
- Gjerga, E., Trairatphisan, P., Gabor, A., Koch, H., Chevalier, C., Ceccarelli, F., Dugourd, A., et al. (2020). Converting networks to predictive logic models from perturbation signalling data with CellNOpt. *Bioinformatics*. doi:[10.1093/bioinformatics/btaa561](https://doi.org/10.1093/bioinformatics/btaa561)
- Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A., Wicks, B., et al. (2012). The Cell Collective: Toward an open and collaborative approach to systems biology. *BMC Systems Biology*, 6(1), 96. doi:[10.1186/1752-0509-6-96](https://doi.org/10.1186/1752-0509-6-96)
- Klarner, H., Streck, A., Siebert, H., & Sahinalp, C. (2016). PyBoolNet: a python package for the generation, analysis and visualization of boolean networks. *Bioinformatics*, 33(5), btw682. doi:[10.1093/bioinformatics/btw682](https://doi.org/10.1093/bioinformatics/btw682)

- Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., Allison, K. R., et al. (2012). Wisdom of crowds for robust gene network inference. *Nature Methods*, 9(8), 796–804. doi:[10.1038/nmeth.2016](https://doi.org/10.1038/nmeth.2016)
- Morris, M. K., Saez-Rodriguez, J., Sorger, P. K., & Lauffenburger, D. A. (2010). Logic-based models for the analysis of cell signaling networks. *Biochemistry*, 49(15), 3216–3224. doi:[10.1021/bi902202q](https://doi.org/10.1021/bi902202q)
- Müssel, C., Hopfensitz, M., & Kestler, H. A. (2010). BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, 26(10), 1378–1380. doi:[10.1093/bioinformatics/btq124](https://doi.org/10.1093/bioinformatics/btq124)
- Naldi, A. (2018). BioLQM: A Java Toolkit for the Manipulation and Conversion of Logical Qualitative Models of Biological Networks. *Frontiers in Physiology*, 9, 1605. doi:[10.3389/fphys.2018.01605](https://doi.org/10.3389/fphys.2018.01605)
- Naldi, A., Hernandez, C., Abou-Jaoudé, W., Monteiro, P. T., Chaouiya, C., & Thieffry, D. (2018). Logical Modeling and Analysis of Cellular Regulatory Networks With GINsim 3.0. *Frontiers in Physiology*, 9, 646. doi:[10.3389/fphys.2018.00646](https://doi.org/10.3389/fphys.2018.00646)
- Naldi, A., Hernandez, C., Levy, N., Stoll, G., Monteiro, P. T., Chaouiya, C., Helikar, T., et al. (2018). The CoLoMoTo Interactive Notebook: Accessible and Reproducible Computational Analyses for Qualitative Biological Networks. *Frontiers in Physiology*, 9, 680. doi:[10.3389/fphys.2018.00680](https://doi.org/10.3389/fphys.2018.00680)
- Naldi, A., Monteiro, P. T., Mussel, C., Kestler, H. A., Thieffry, D., Xenarios, I., Saez-Rodriguez, J., et al. (2015). Cooperative development of logical modelling standards and tools with CoLoMoTo. *Bioinformatics*, 31(7), 1154–1159. doi:[10.1093/bioinformatics/btv013](https://doi.org/10.1093/bioinformatics/btv013)
- Paulevé, L. (2017). Pint: A Static Analyzer for Transient Dynamics of Qualitative Networks with IPython Interface. In *CMSB 2017 - 15th conference on computational methods for systems biology*, Lecture notes in computer science (Vol. 10545, pp. 316–370). Springer. doi:[10.1007/978-3-319-67471-1_20](https://doi.org/10.1007/978-3-319-67471-1_20)
- Senft, D., Leiserson, M. D. M., Ruppin, E., & Ronai, Z. A. (2017). Precision Oncology: The Road Ahead. *Trends in Molecular Medicine*, 23(10), 874–898. doi:[10.1016/j.molmed.2017.08.003](https://doi.org/10.1016/j.molmed.2017.08.003)
- Stoll, G., Caron, B., Viara, E., Dugourd, A., Zinovyev, A., Naldi, A., Kroemer, G., et al. (2017). MaBoSS 2.0: an environment for stochastic Boolean modeling. (J. Wren, Ed.) *Bioinformatics*, 33(14), 2226–2228. doi:[10.1093/bioinformatics/btx123](https://doi.org/10.1093/bioinformatics/btx123)
- Terfve, C., Cokelaer, T., Henriques, D., MacNamara, A., Goncalves, E., Morris, M. K., Iersel, M. van, et al. (2012). CellINOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Systems Biology*, 6(1), 133. doi:[10.1186/1752-0509-6-133](https://doi.org/10.1186/1752-0509-6-133)
- Traynard, P., Tobalina, L., Eduati, F., Calzone, L., & Saez-Rodriguez, J. (2017). Logic Modeling in Quantitative Systems Pharmacology. *CPT: Pharmacometrics & Systems Pharmacology*, 6(8), 499–511. doi:[10.1002/psp4.12225](https://doi.org/10.1002/psp4.12225)
- Wang, R.-S., Saadatpour, A., & Albert, R. (2012). Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5), 55001. doi:[10.1088/1478-3975/9/5/055001](https://doi.org/10.1088/1478-3975/9/5/055001)
- Zobolas, J. (2020a). DrugLogics software documentation. GitHub Pages. Retrieved from <https://druglogics.github.io/druglogics-doc/>
- Zobolas, J. (2020b). Usefun: A collection of useful functions by john. GitHub Pages. Retrieved from <https://github.com/bblodfon/usefun>

Zobolas, J. (2020c). Emba package website. GitHub Pages. Retrieved from <https://bblodfon.github.io/emba/>

Zobolas, J. (2020d). Ensemble boolean model analyses related to drug prediction performance. GitHub. Retrieved from <https://github.com/bblodfon/gitsbe-model-analysis>

PAPER 5

BOOLEAN FUNCTION METRICS CAN ASSIST MODELERS TO CHECK AND CHOOSE LOGICAL RULES

John Zobolas^{1,*}, Pedro T. Monteiro^{2,3}, Martin Kuiper¹ and Åsmund Flobak^{4,5}

¹Department of Biology, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

²Department of Computer Science and Engineering, Instituto Superior Técnico (IST) - Universidade de Lisboa, Lisbon, Portugal

³INESC-ID, Lisbon, Portugal

⁴Department of Clinical and Molecular Medicine, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

⁵The Cancer Clinic, St. Olav's Hospital, Trondheim, Norway

*To whom correspondence should be addressed.

April 6, 2021

Abstract

Computational models of biological processes provide one of the most powerful methods for a detailed analysis of the mechanisms that drive the behavior of complex systems. Logic-based modeling has enhanced our understanding and interpretation of those systems. Defining rules that determine how the output activity of biological entities is regulated by their respective inputs has proven to be challenging. Partly this is because of the inherent noise in data that allows multiple model parameterizations to fit the experimental observations, but some of it is also due to the fact that models become increasingly larger, making the use of automated tools to assemble the underlying rules indispensable.

We present several Boolean function metrics that provide modelers with the appropriate framework to analyze the impact of a particular model parameterization. We demonstrate the link between a semantic characterization of a Boolean function and its consistency with the model's underlying regulatory structure. We further define the properties that outline such consistency and show that several of the Boolean functions under study violate them, questioning their biological plausibility and subsequent use. We also illustrate that regulatory functions can have major differences with regard to their asymptotic output behavior, with some of them being biased towards specific Boolean outcomes when others are dependent on the ratio between activating and inhibitory regulators.

Application results show that in a specific signaling cancer network, the function bias can be used to guide the choice of logical operators for a model that matches data observations. Moreover, graph analysis indicates that the standardized Boolean function bias becomes more prominent with increasing numbers of regulators, confirming the fact that rule specification can effectively determine regulatory outcome despite the complex dynamics of biological networks.

Keywords Boolean regulatory networks · Boolean functions · Truth Density · Bias · Complexity

1 Introduction

The understanding of biological processes has been greatly stimulated by systems biology approaches [1, 2, 3]. The integration of mathematical models with the underlying biological knowledge and empirical observations can help us observe emergent systems properties, test new hypotheses, enhance the interpretability of the studied systems and guide innovations in areas such as medicine and drug discovery [4]. While multiple mathematical modeling frameworks exist, the scarcity of experimental data and the challenges posed by the development of quantitative large-scale biological networks, has favoured the simplicity and intuitiveness of more qualitative approaches, such as logic-based modeling [5].

At the heart of the mathematical representation of molecular biological networks lies the concept of regulation. Regulation of activity, typically by changing the modification state, location or concentration of a biological entity, is a process which can be expressed by a mathematical function that combines the various regulatory inputs that affect the target, with a logic that describes how these regulators are integrated. In Boolean logic-based modeling, the regulatory inputs are entities which can be expressed in two states: active (1) or inactive (0). These entities are combined with logical rules to derive the *Boolean regulatory function* (BRF) of the target entity. For every possible regulatory input (combination of 0 and 1's) the BRF will produce the end regulatory product, which is the activity of the target (0 or 1).

The construction of a Boolean computational model starts with the assembly of information from literature and experimental observations, in the form of a Prior Knowledge Network (PKN), i.e. a list of network entities and their causal interactions (positive or negative) [6, 7]. The use of a PKN for accurate representation of biological reality and subsequent analysis and simulation requires the definition of the model formalism. This is one of the most important steps in dynamical modeling since it directly translates to the choice of BRFs, i.e. the logical rules that together with the regulators define the activity state of each network target [8]. There have been several approaches related to the choice of BRFs, from using a standardized format [9], to automatically generating all possible BRFs compatible with the PKN and calibrating the rules in order to fit perturbation data [10, 11, 12]. State-of-the-art approaches involve the automated construction of large-scale logical networks by inferring the logical rules from the topology and semantics of molecular interaction maps [13].

Regardless of how a logical model is constructed, it has been shown in practice that expert curation, i.e. the manual fine-tuning of the logical rules to fit experimental data, can result in highly predictive models [14, 15], yet this is not trivially obtained with automatically constructed networks [16]. Because of the large function space complemented with a sparsity of observations and inherent noise in existing data, there is a wide range of plausible BRFs. Thus, it is crucial to properly define function characteristics that can guide the modeler to a more informed function choice. Our work is focused on explicating some of these metrics and using them to show for example which BRFs can be discarded due to biological inconsistencies with the underlying regulatory topology and which are biased towards specific Boolean outcomes.

The paper is structured as follows: Section 2 provides a list of notations and definitions to be used later in the text. In Section 3, we discuss the benefits of using the equivalent disjunctive normal form of a Boolean function to delineate its biological interpretability. In Section 4, we provide a set of properties that characterize the Boolean functions that are consistent with a given regulatory topology and show that several functions under study violate them. In Section 5, we present the truth density metric as a means to evaluate if a Boolean function is biased or balanced with increasing number of regulators. We also discuss the asymptotic properties of different functions relating to the ratio between activators and inhibitors. Lastly, in Section 6, we present evidence that the standardized Boolean functions are indeed biased and show how modelers can exploit such information for their own benefit. The results are demonstrated in Boolean models derived from a cancer signaling network as well as from scale-free topologies that are applicable to most biological networks. We close the paper with some discussion points in Section 7 and directions for future research in Section 8.

2 Background

2.1 Boolean regulatory functions

Boolean regulatory functions (BRFs) are Boolean functions used in the context of biological networks and modeling. A mathematical description of such a function associates the activity output of a target biological entity with the Boolean input values of n variables (the *regulators*), such that $f_{BRF} : \{0, 1\}^n \rightarrow \{0, 1\}$. Thus, the target's output state is binary, i.e. either 0 (*False*, denoting an inactive or inhibited state) or 1 (*True*, indicating an active state).

One intuitive representation of a Boolean function is its *truth table*, which is a list of all possible Boolean input configurations of the n regulators along with their associated function output. Since every regulator can be assigned two possible values (0 and 1), the total number of input configurations (i.e. rows) in a truth table is 2^n . For example, a Boolean function $f(x_1, x_2, x_3)$ with 3 regulators has a total of $2^3 = 8$ rows in its corresponding truth table, starting from the input configuration $(0, 0, 0)$ and ending with $(1, 1, 1)$ (Table 1).

The total number of BRFs with n regulators is 2^{2^n} since for each of the 2^n input configurations (i.e. rows of the truth table) there can be two possible function outcomes (0 or 1). For example, with 3 regulators and a total of 8 rows in the truth table, that would be a total of $2^8 = 256$ functions, three of which are shown in Table 1.

2.2 Disjunctive normal form

The most frequently used form of a Boolean function is its analytical expression, where variables are connected with logical operators such as AND (\wedge), OR (\vee), NOT (\neg), XOR (\oplus), etc. and the output of the function is calculated using basic Boolean algebra. In Table 1 for example, we provide the analytical forms for the functions f_1 and f_2 . Note that there can be multiple analytical forms that essentially compute the same function, e.g. another form of the f_1 function is $f'_1 = (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge \neg x_3)$.

Truth Table			Boolean functions		
x_1	x_2	x_3	$f_1 = (x_1 \wedge \neg x_3) \vee (x_2 \wedge \neg x_3)$	$f_2 = x_1 \vee (\neg x_2 \wedge \neg x_3)$	$f_3 = 1$
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1

Table 1: Truth table of three Boolean functions with three input variables x_1, x_2 and x_3 . Functions f_1 and f_2 are expressed in disjunctive normal form (DNF) with the minimum possible number of terms. f_3 is a tautology.

This brings us to the notion of a general form which could be used to define useful metrics common to all Boolean functions (e.g. complexity), as well as the need to provide minimal forms based on specific criteria. For example, a more compact function form enhances readability, which can be seen by comparing f_1 with f'_1 .

Every Boolean function can be represented in a *disjunctive normal form* (DNF), requiring only AND (\wedge), OR (\vee) and NOT (\neg) operators as building blocks. In such a representation, *literals*, which are variables (e.g. positive literal x) or their logical negations (e.g. negative literal NOT x), are connected by AND's, producing *terms*, which are then in turn connected by OR's [17]. For example, every function in Table 1 is expressed in DNF, while the Boolean expressions $\neg(x_1 \vee x_2)$ and $\neg(x_1 \wedge x_2) \vee x_3$ are not. Note that a Boolean function can have multiple DNF formulations.

2.3 Link operator functions

We consider the class of BRFs that partitions the input regulators to two sets: the set of positive regulators (*activators*) and the set of negative regulators (*inhibitors*). Let f be such a Boolean function $f_{BRF}(x, y) : \{0, 1\}^n \rightarrow \{0, 1\}$, with $m \geq 1$ activators $x = \{x_i\}_{i=1}^m$ and $k \geq 1$ inhibitors $y = \{y_j\}_{j=1}^k$, that is a total of $n = m + k$ regulators. The *link operator* BRFs have an analytical formula which places the two distinct types of regulators in two separate expressions, while connecting them with a special logical operator that we call a *link operator*. An example of such a function that has been used extensively in the logical modeling literature is the standardized BRF formula with the “AND-NOT” link operator [9]:

$$f_{AND-NOT}(x, y) = \left(\bigvee_{i=1}^m x_i \right) \wedge \neg \left(\bigvee_{j=1}^k y_j \right) \quad (1)$$

A variation of the above function is the “OR-NOT” link operator function:

$$f_{OR-NOT}(x, y) = \left(\bigvee_{i=1}^m x_i \right) \vee \neg \left(\bigvee_{j=1}^k y_j \right) \quad (2)$$

Note that the presence of the link operator is what forces the condition $m, k \geq 1$ (at least one regulator in each category). For the rest of this work, we will not consider BRFs with only one type of regulator, since these can be represented by simple logical functions without loss of biological consistency. Following the notation introduced in Mendoza et al. [9], in the case of only positive regulators, the presence of at least one activator makes the target active, i.e. $f(x) = \bigvee_{i=1}^m x_i$. In the case of only inhibitory regulators, the presence of at least one inhibitor is sufficient to make the target inactive, i.e. $f(y) = \neg \bigvee_{j=1}^k y_j = \bigwedge_{j=1}^k \neg y_j$.

Borrowing notation from circuit theory, we will also use other link operators like the “NAND”, “NOR”, “XNOR” gates, with or without the “NOT” symbol in front. Note that the logical operator used to connect the same type of regulators (e.g. the activators) is usually OR, but other operators could be used as well.

Another link operator function that we will consider in this work is the “Pairs” function:

$$f_{Pairs}(x, y) = \bigvee_{\forall(i,j)}^{m,k} (x_i \wedge \neg y_j) \quad (3)$$

The intuition behind the name is derived from the fact that the function will return *True* if there is at least one pair of regulators consisting of a present activator and an absent inhibitor. For a formulation of the “Pairs” function that is consistent with the link operator terminology as defined above, see (Eq. 9).

2.4 Threshold functions

Threshold functions are a special type of Boolean functions, the output of which depends on the condition that the sum of (possibly weighted) activities of the input regulators surpasses a given *threshold* value [18, 19].

In this work we will consider two simple threshold functions, which both output *True* when the number of present activators is larger than the number of present inhibitors. As such, the activities of the positive and negative regulators are combined in an *additive* manner, with their respective assigned weights set to ± 1 and the threshold parameter to 0, formulating thus a *majority rule* which defines the value of the function [20, 21]. These functions differ with regards to their output when there is balance between the activities of the positive and negative regulators: the first outputs 1 (the activators “win”) while the second outputs 0 (the inhibitors “win”):

$$f_{Act-win}(x, y) = \begin{cases} 1, & \sum_{i=1}^m x_i \geq \sum_{j=1}^k y_j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$f_{Inh-win}(x, y) = \begin{cases} 1, & \sum_{i=1}^m x_i > \sum_{j=1}^k y_j \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

3 Disjunctive Normal Form unmasks biological interpretation

3.1 Interpretability issues in Boolean modeling

Two main features make Boolean modeling attractive to users. First, transforming conditions for the activation or inhibition of a target biological entity to Boolean equations is a relatively easy task using a qualitative, logic-based modeling formalism. Second, the reverse is also true, i.e. Boolean equations can be more interpretable and closer to a simplified description of biological reality that “makes sense” than the use of other kinds of formalisms (e.g. kinetic modeling). For example, consider the simple case of a target entity, which is regulated by one positive regulator x_1 and one negative regulator y_1 . The use of the “AND-NOT” link operator function in this case (Eq. 1) is very easy to understand and interpret since the formula directly connects to the underlying biology. Thus, the mathematical formulation is simply written as $f_{AND-NOT} = x_1 \text{ AND NOT } y_1$, while the modeler reads “the target becomes active when x_1 (the activator) is present and y_1 (the inhibitor) absent”.

Issues start arising when considering the *interpretability* of such Boolean expressions in cases where a larger number of regulators act on a target, e.g. in a more complex scenario with three positive (x_1, x_2, x_3) and three negative (y_1, y_2, y_3) regulators, the mathematical formulation expressing the target’s activity output can be easily written using the link operator function form, as $f_{AND-NOT} = (x_1 \text{ OR } x_2 \text{ OR } x_3) \text{ AND NOT } (y_1 \text{ OR } y_2 \text{ OR } y_3)$. A modeler could read this as “the target becomes active when at least one activator is present, and all of its inhibitory regulators are absent”, but a precise semantic description that explicates the conditions under which the target gets activated, can in general be difficult to assess. A similar issue arises when reflecting on the use of a different link operator instead of the standard “AND-NOT” or even of an entirely different regulatory function, for which the biological interpretation might be difficult to derive from the expression itself.

BRF (standard form)	BRF (CDNF)	Biological Interpretation	Consistent	Complexity
$(x_1 \text{ OR } x_2) \text{ NOR } (y_1 \text{ OR } y_2)$	$\text{NOT } x_1 \text{ AND NOT } x_2 \text{ AND NOT } y_1 \text{ AND NOT } y_2$	Absence of all regulators	NO	1 (always)
$(x_1 \text{ OR } x_2) \text{ NAND } (y_1 \text{ OR } y_2)$	$(\text{NOT } x_1 \text{ AND NOT } x_2) \text{ OR } (\text{NOT } y_1 \text{ AND NOT } y_2)$	Absence of all activators or absence of all inhibitors	NO	2 (always)
$(x_1 \text{ OR } x_2) \text{ AND NOT } (y_1 \text{ OR } y_2)$ “AND-NOT” (Eq. 1)	$(x_1 \text{ AND NOT } y_1 \text{ AND NOT } y_2) \text{ OR } (x_2 \text{ AND NOT } y_1 \text{ AND NOT } y_2)$	Presence of at least one activator and absence of all inhibitors	YES	2 (m)
$(x_1 \text{ OR } x_2) \text{ NOR NOT } (y_1 \text{ OR } y_2)$	$(y_1 \text{ AND NOT } x_1 \text{ AND NOT } x_2) \text{ OR } (y_2 \text{ AND NOT } x_1 \text{ AND NOT } x_2)$	Presence of at least one inhibitor and absence of all activators	NO	2 (k)
$(x_1 \text{ OR } x_2) \text{ OR NOT } (y_1 \text{ OR } y_2)$ “OR-NOT” (Eq. 2)	$x_1 \text{ OR } x_2 \text{ OR } (\text{NOT } y_1 \text{ AND NOT } y_2)$	Presence of any activator or absence of all inhibitors	YES	3 ($m + 1$)
$(x_1 \text{ OR } x_2) \text{ NAND NOT } (y_1 \text{ OR } y_2)$	$y_1 \text{ OR } y_2 \text{ OR } (\text{NOT } x_1 \text{ AND NOT } x_2)$	Presence of any inhibitor or absence of all activators	NO	3 ($k + 1$)
$(x_1 \text{ OR } x_2) \text{ XOR } (y_1 \text{ OR } y_2)$	$(x_1 \text{ AND NOT } y_1 \text{ AND NOT } y_2) \text{ OR } (x_2 \text{ AND NOT } y_1 \text{ AND NOT } y_2) \text{ OR } (\text{NOT } x_1 \text{ AND NOT } x_2 \text{ AND } y_1) \text{ OR } (\text{NOT } x_1 \text{ AND NOT } x_2 \text{ AND } y_2)$	Presence of at least one activator and absence of all inhibitors or presence of at least one inhibitor and absence of all activators	NO	4 ($m + k$)
$(x_1 \text{ OR } x_2) \text{ AND } (\text{NOT } y_1 \text{ OR NOT } y_2)$ “Pairs” (Eq. 3)	$(x_1 \text{ AND NOT } y_1) \text{ OR } (x_1 \text{ AND NOT } y_2) \text{ OR } (x_2 \text{ AND NOT } y_1) \text{ OR } (x_2 \text{ AND NOT } y_2)$	Presence of at least one activator and absence of at least one inhibitor	YES	4 ($m \times k$)
$(x_1 \text{ OR } x_2) \text{ XNOR } (y_1 \text{ OR } y_2)$	$(x_1 \text{ AND } y_1) \text{ OR } (x_1 \text{ AND } y_2) \text{ OR } (x_2 \text{ AND } y_1) \text{ OR } (x_2 \text{ AND } y_2) \text{ OR } (\text{NOT } x_1 \text{ AND NOT } x_2 \text{ AND NOT } y_1 \text{ AND NOT } y_2)$	Presence of at least one activator and inhibitor pair or absence of all regulators	NO	5 ($m \times k + 1$)
<i>True</i> when $x_1 + x_2 > y_1 + y_2$ “Inh-win” (Eq. 5)	$(x_1 \text{ AND } x_2 \text{ AND NOT } y_1) \text{ OR } (x_1 \text{ AND } x_2 \text{ AND NOT } y_2) \text{ OR } (x_1 \text{ AND NOT } y_1 \text{ AND NOT } y_2) \text{ OR } (x_2 \text{ AND NOT } y_1 \text{ AND NOT } y_2)$	Number of present activators is larger than the number of present inhibitors	YES	4
<i>True</i> when $x_1 + x_2 \geq y_1 + y_2$ “Act-win” (Eq. 4)	$(x_1 \text{ AND } x_2) \text{ OR } (x_1 \text{ AND NOT } y_1) \text{ OR } (x_1 \text{ AND NOT } y_2) \text{ OR } (x_2 \text{ AND NOT } y_1) \text{ OR } (x_2 \text{ AND NOT } y_2)$	Number of present activators is larger than or equal to the number of present inhibitors	YES	5

Table 2: Several Boolean regulatory functions with four regulators ($m = 2$ positive $\{x_1, x_2\}$, $k = 2$ negative $\{y_1, y_2\}$) and some metrics are presented. The two first columns provide two different function forms: a standard one, i.e. either the link operator form distinguishing activating and inhibiting regulators or a simple description in the case of the threshold functions, and the CDNF which is a special case of DNF (Section 4.1). The “Biological Interpretation” states in words the conditions that make a BRF become *True*, and is explicitly translated from the terms in the corresponding CDNF. The “Consistent” column states if the functions satisfy the properties 1-3 from Section 4.1 (YES, green-colored) or there are inconsistencies with the underlying regulatory structure (NO, red-colored), i.e. if an activator (resp. inhibitor) appears as a negative (resp. positive) literal in the corresponding CDNF. The functions are sorted according to an increasing complexity metric (“Complexity” column), which is the number of terms in each respective, minimum-length CDNF expression. In parentheses we provide the generalized formula for the number of CDNF terms of the link operator functions with m activators and k inhibitors.

3.2 DNF links to biological semantics

We argue here that the DNF is the most adequate function form to help us address the aforementioned issues. Every Boolean regulatory function expressed in DNF, has a biological characterization that is directly derived from the formula itself: each term in the DNF is an activation condition, i.e. a list of regulators, some present (the positive literals) and some absent (the negative literals), which, when combined, make the target (output of the function) active. Further merging of all the conditions using OR-semantics into a description of how the regulators influence the target’s output, facilitates the biological interpretation of any Boolean regulatory function.

In Table 2, we show a list of BRFs with two positive and two negative regulators. Most of the BRFs presented have a different link operator separating the activators from the inhibitors. Using the functions standard expressions (1st column) makes it very hard to derive a meaningful biological characterization as expressed in the 3rd column of Table 2. For example, defining a meaningful description of the “NOR” or “NAND-NOT” equations using only their standard expression, is a very difficult task. In contrast, by using the equivalent DNFs (2nd column) we can make an explicit, “1-1” correspondence between mathematical formulation and biological interpretation and use it to compare the different functions’ meanings. Thus, by expressing the “NAND-NOT” equation in DNF, we can precisely identify the conditions that make the outcome of the function *True* and translate these into a meaningful description such as “Presence of any inhibitor or absence of all activators”. Consequently, we are led to a generalized and independent of the number of regulators description of this link operator function. Such a description is intuitive to human interpretation and reasoning, in terms of the function’s applicability, e.g. in comparing the “AND-NOT” and “NAND-NOT” biological interpretations, we see that the first is semantically plausible while the second completely contradicts the underlying biology.

4 Characterizing consistent regulatory functions

4.1 The 3 consistency properties

As observed in Table 2, not only can the DNF be used to uncover the biological interpretation of any BRF and subsequently help determine its plausibility, but it also provides a means to compare the different function meanings. Still, we need a more refined, technical description that is able to express the implausibility of the “NAND-NOT” or “NOR” cases directly from their mathematical formulas, and which would be applicable to every BRF. We define the *consistency* attribute of a BRF to describe its compliance with the underlying regulatory network structure.

The first step in making a Boolean model is to build a graph (PKN), assembling the regulatory entities of interest from various databases or the scientific literature, and use causality information to connect them through their regulatory action on other entities. As such, a network structure can be defined, in which entities can regulate (either positively or negatively) some of the other entities. Using such a simple network-driven formalization, we define a set of three properties that describe the set of all the *consistent Boolean regulatory functions*, i.e. the functions that comply with the underlying regulatory structure. So, for a consistent BRF, the following propositions are satisfied [22]:

1. Its regulators can be partitioned into two disjoint sets: the set of *activators* (positive regulators, enhance target’s activity) and the set of *inhibitors* (negative regulators, suppress target’s activity). This stems from the fact that every interaction in the PKN has a fixed sign (either positive or negative). As such, there are no dual regulations, i.e. a regulator cannot activate and inhibit a target at the same time. This property essentially makes the set of consistent BRFs a subset of the *monotone* Boolean functions [17].

2. All regulators are *essential*: for every regulatory input, inverting their values, will also, in at least one configuration of states of other regulators, change the output of the function. This means that all regulators are indispensable for deriving the target’s activity output.
3. A consistent BRF can be represented in a unique *complete* DNF (CDNF) which is also known as Blake’s Canonical Form [23]. This is a consequence of property (1), since monotone Boolean functions expressed in any DNF, can be further simplified by removing redundant literals, resulting in the equivalent unique CDNF expression [17]. This property is really important since it allows us to identify which regulatory entities are activators and which are inhibitors from the corresponding CDNF expression of a consistent BRF: an activator will always appear as a positive literal, whereas an inhibitor will always appear as a negative literal.

We provide an example to delineate the difference between the DNF and CDNF forms and show violations of the consistency properties. In Table 3, we present three Boolean functions, expressing the output of a target regulated by one activator (x_1) and one inhibitor (y_1). The functions f_2 and f_3 are in CDNF whereas f_1 is in normal DNF, since the positive regulator x_1 appears both as a positive and a negative literal (i.e. it acts as a dual regulator, making f_1 inconsistent). Notice that f_1 reduces to f_2 by removing the redundant negative literal ($\neg x_1$) in the term ($\neg x_1 \wedge y_1$): y_1 “absorbs” the larger term and thus a shorter expression manifests, one that covers more *True* outcomes (i.e. 1’s) in the truth table. In addition, we observe that f_2 is inconsistent, since inhibitor y_1 appears as a positive literal. On the other hand, using a negative literal for inhibitor y_1 and a positive one for activator x_1 , makes f_3 consistent.

Truth Table		Term	Boolean functions		
x_1	y_1	$(\neg x_1 \wedge y_1)$	$f_1 = x_1 \vee (\neg x_1 \wedge y_1)$	$f_2 = x_1 \vee y_1$	$f_3 = x_1 \vee \neg y_1$
0	0	0	0	0	1
0	1	1	1	1	0
1	0	0	1	1	1
1	1	0	1	1	1

Table 3: Truth table of three different Boolean regulatory functions with two input regulators, one positive (x_1) and one negative (y_1). All functions are expressed in DNF. f_1 and f_2 result in the same target Boolean output, with f_2 expressed in CDNF. Activator x_1 regulates the target both positively and negatively in f_1 , making the function non-monotone and thus inconsistent. Inhibitor y_1 is a positive literal in f_2 ’s CDNF, making it inconsistent as well. Function f_3 is consistent since it’s written in CDNF with the activator x_1 and inhibitor y_1 appearing as positive and negative literals respectively.

4.2 Most link operator functions are inconsistent

Examining Table 2, we note that the 2nd column presents not just any DNF expression of the studied Boolean regulatory functions, but precisely the CDNF. Thus we can immediately identify which BRFs violate at least one of the three properties discussed in Section 4.1 and are therefore inconsistent with the regulatory topology (this information is presented in the 4th column, labeled “Consistent”). Two examples of such inconsistencies include the “NAND-NOT” and “XOR” link operator functions, which have terms in their corresponding CDNF in which an activator x_i appears as a negative literal (NOT x_i) and an inhibitor y_j as a positive literal (as itself). In total, from all the BRFs presented in Table 2, only the standardized “AND-NOT”, the “OR-NOT”, the “Pairs” and the two threshold functions respect the underlying regulatory topology, as

can be verified by examining their respective CDNFs. The rest of the link operator functions presented are inconsistent and will not be considered for further analysis in this paper.

5 Truth Density as a measure of expected function output

In this section we present another interesting Boolean function metric, whose properties can be used to add further knowledge about a Boolean function's behavior. This metric, which we call *truth density*, allows us to project what the regulatory target's output will most likely be when the number of input regulators changes and investigate how the ratio between activators and inhibitors may affect that output. From a modeler's perspective, this metric is useful to check if an assigned model parameterization (i.e. use of a specific BRF) can asymptotically predefine the activity state of some targets. Equipped with this knowledge, a modeler can verify the degree of fitness with the observations that such a parameterization allows, and thus discard a specific function in favor of another, if the latter has a truth density value that better matches the outcome observed in the data.

5.1 Truth Density

We define the truth density (TD) of a Boolean function as the fraction of all input configurations in its corresponding truth table that yield a *True* (1) outcome. As such, $TD \in [0, 1]$. This quantity was first introduced in [24] and more recently in [25] under the name of *bias* and was similarly defined as the probability that a Boolean function takes on the value 1. Using the example with the three Boolean functions from Table 1, we have $TD_{f_1} = 3/8 = 0.375$, $TD_{f_2} = 5/8 = 0.625$ and $TD_{f_3} = 8/8 = 1$, where the last function is a tautology, with the maximum possible truth density. Colloquially, we can say that a Boolean function is *biased*, when it's truth density is close to 0 or 1. Since the size of a truth table grows exponentially with the number of inputs of the Boolean function (n inputs correspond to 2^n rows), the existence of bias conveys the information that most of the input regulatory configurations result in either an activated or inhibited target (bias towards 1 or 0 respectively). On the other hand, we shall say that a Boolean function is *balanced*, if it takes on the values 0 and 1 equally often, or equivalently, it's truth density is approximately centered around 1/2 [26].

5.2 Asymptotic truth density results of the consistent regulatory functions

In Appendix A, we present a list of propositions and proofs that provide the exact truth density formulas for the generic forms of the five consistent BRFs we studied in previous sections, namely the “AND-NOT”, “OR-NOT” and “Pairs” link operator functions, and the two threshold functions, “Act-win” and “Inh-win”. A very important element that enables the straightforward derivation of these formulas, is the use of the equivalent DNF expressions in the proofs, especially for the case of the link operator Boolean functions. We also noted that the truth densities of all the aforementioned BRFs depend on two variables: the number of activators and the number of inhibitors (the total number of regulators also appears as a separate variable but it depends on the first two, i.e. it is just their sum). Thus, we logically asked if a BRF's truth density asymptotically tends towards specific values in the $[0, 1]$ interval (e.g. the function could be biased or balanced), when the number of its input regulators increases or the ratio between activators and inhibitors changes. The results of the asymptotic behavior of the truth density formulas are analytically presented in Appendix B.

The asymptotic analysis of the truth density formulas confirmed the intuitive perception that the link operator “AND-NOT” and “OR-NOT” functions show a characteristically opposite behavior with increasing number of regulators: the standardized “AND-NOT” formula depends only on the number of inhibitors and its output tends towards 0, whereas the “OR-NOT” formula depends only on the number of activators and is biased towards 1. On the other hand, the “Pairs” and threshold functions truth densities don't have an asymptotic

limit since they depend on both the number of activators and inhibitors. Therefore, we proceeded in clarifying the role of the *activator-to-inhibitor* ratio by investigating three scenarios which explicitly reveal the functions truth density behavior for a significantly large number of regulators:

- A 1 : 1 activator-to-inhibitor ratio, where approximately half of the regulators are activators and half are inhibitors.
- A high activator-to-inhibitor ratio, where all regulators are activators except one inhibitor.
- A low activator-to-inhibitor ratio, where all regulators are inhibitors except one activator.

In the 1 : 1 ratio scenario, where there is an equal number of activators and inhibitors, the asymptotic behavior of the “AND-NOT” and “OR-NOT” functions corresponds to absolute inhibition (0) and activation (1) respectively, following the biased behavior shown previously. The “Pairs” function behaves similarly to the “OR-NOT” function and therefore is also biased towards 1. Only the threshold functions show balanced behavior with their truth density value reaching asymptotically 1/2, since the majority rule does favor neither activators nor inhibitors in this scenario. On the other hand, in the two extremely unbalanced scenarios, where one set of regulators completely outweighs the other, the asymptotic truth density results of the “AND-NOT” and “OR-NOT” functions depend on each respective scenario. Specifically, when the inhibitors dominate over the activators, the “OR-NOT” is balanced and the “AND-NOT” is biased, since the former has been shown to depend exclusively on the number of activators (which is just one in this case) for increasingly more regulators, whereas the latter on the number of inhibitors. Their behavior is reversed when the activators outbalance the inhibitors. In contrast, the “Pairs” function behaves in a balanced manner, having a truth density asymptotically equal to 1/2 in both these scenarios, since the single minority regulator is paired with every regulator from the dominant group in the respective DNF expression (Eq. 3) and as a result, it significantly influences the function’s output. Lastly, the asymptotic results for the threshold functions follow the larger size regulatory group, being biased towards 0 with significantly more inhibitors and biased towards 1 with significantly more activators.

5.3 Validation of asymptotic behavior

One key issue of immense practical importance for the modeler, which arises when analyzing the asymptotic behavior of the truth density formulas, is the actual number of regulators that effectively make each of the studied functions exhibit the demonstrated behavior. We noticed that most of the truth density formulas (Eq. 11, 12 and 13) are the sum of two to three terms, with only one of them depending exclusively on the number of regulators n . Also, this term is usually $1/2^n$, and can be omitted when considering values larger than $n = 10$ regulators since it’s insignificant ($1/2^{10} \approx 0.001$). This suggests that the limit value of the truth density formulas may be already derived from a much smaller number of regulators than what is implied by the study of asymptotes. Therefore, we need to have a more data-centric view of the results from our previous asymptotics analysis of the different BRFs, one that will enable us to verify the mathematically observed behaviors but also identify an approximate range for the number of regulators where the asymptotics decide the outcome of the studied functions.

We generated the complete truth tables for the five consistent BRFs of Table 2, from 2 up to 20 regulators, accounting for every possible activator-to-inhibitor ratio. For example, for $n = 10$ regulators, every combination of at least one activator and one inhibitor that adds up to 10 (1 activator + 9 inhibitors, 2 activators + 8 inhibitors, etc.) resulted in a different truth table for each considered Boolean function. Subsequently, using the generated truth tables, we could easily calculate the exact truth density value for each function at every considered ratio. The results are shown in Figures 1A and 1B for the link operator and threshold functions, respectively.

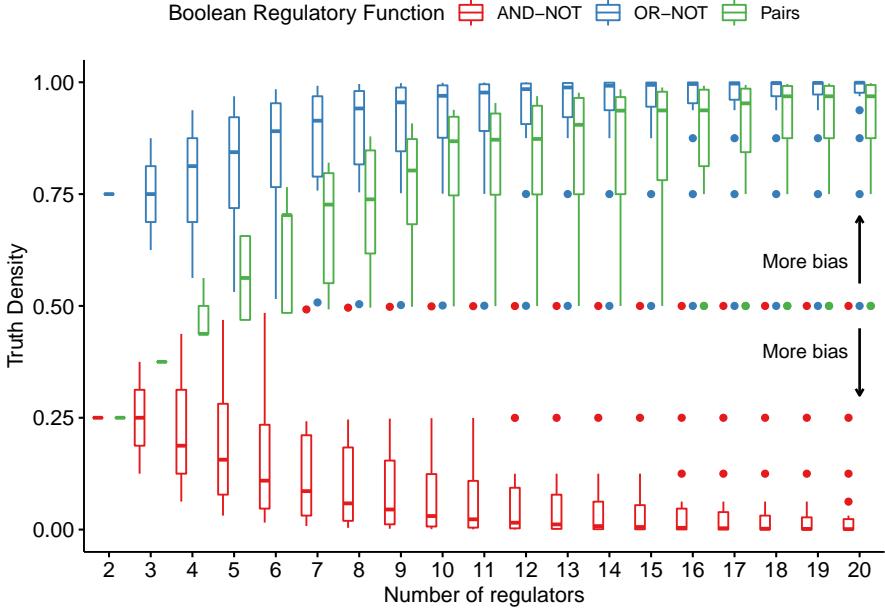
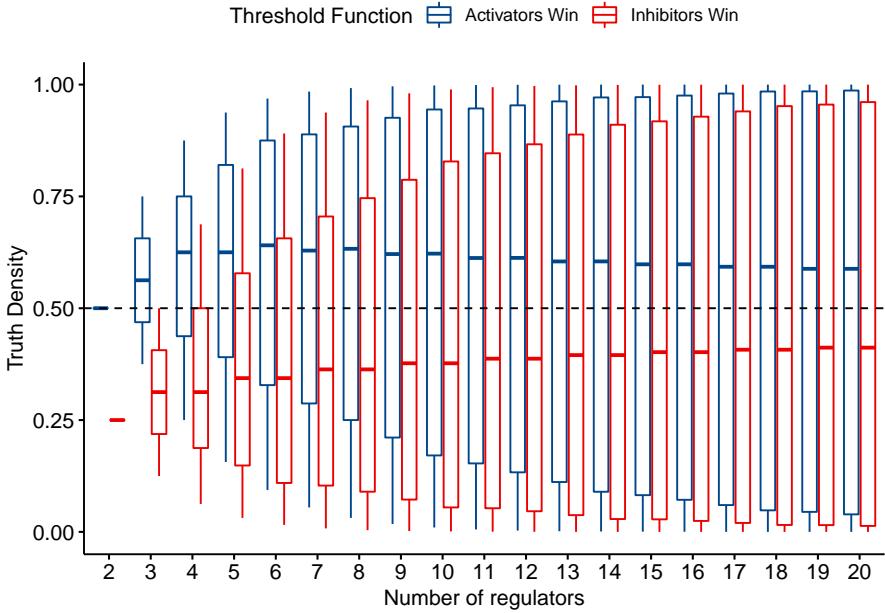
A**B**

Figure 1: Comparing the truth densities of five different Boolean regulatory functions for different numbers of regulators and activator-to-inhibitor ratios. For each specific number of regulators, every possible combination of at least one activator and one inhibitor that add up to that number, results in a different truth table output with its corresponding truth density value. All such possible configurations up to 20 regulators are shown. **(A)** The standardized “AND-NOT” function, along with the “OR-NOT” and “Pairs” functions, show an increasingly biased behavior with more regulators. **(B)** The two threshold functions “Act-win” and “Inh-win” show a more balanced behavior, since they respect the activator-to-inhibitor ratio and thus demonstrate a larger spectrum of possible truth density values even for higher numbers of regulators.

The data in general shows that the different regulatory functions demonstrate quite dissimilar behaviors with regard to their asymptotic outcome. In particular, we recapitulate the findings from the asymptotics analysis, namely the bias of the link operator functions, which is evident even from 7 to 10 input regulators. Interestingly, the “Pairs” function follows asymptotically the behavior of the “OR-NOT” function but is in general less biased. We note that the outliers in Figure 1A with truth density values closer to 1/2, represent imbalanced activator-to-inhibitor ratio scenarios, i.e. either considerably more activators than inhibitors for the “AND-NOT” function and the reverse for the “OR-NOT” function, or any imbalanced ratio for the “Pairs” function. Lastly, Figure 1B shows that the threshold functions exhibit a more balanced behavior, expressed as a higher spectrum of truth density values for any single number of regulators and with the median truth density asymptotically reaching 1/2. This result is due to the fact that threshold functions faithfully follow the activator-to-inhibitor ratio, i.e. with more activators the outcome is biased towards 1 whereas with more inhibitors the function outcome tends towards 0.

6 Link operator parameterization determines activity state in biological networks

In this section we investigate if a model’s parameterization can effectively decide the activity state of nodes in biological networks. In more detail, we will use the “AND-NOT” link operator function [9] and its symmetric function “OR-NOT” (Eq. 1 and 2), to build Boolean models from prior causal knowledge and check if their activity state profile as determined by dynamic attractor analysis, shows the biased behavior that we observed in Section 5.

A major motivation for this analysis is the fact that the “AND-NOT” function is extensively used by logical modelers and thus the knowledge of its bias, made possible through the lens of the truth density metric, should be clearly demonstrated in practical use cases, e.g. biological network targets should mostly be in an inhibited state when the “AND-NOT” parameterization is used in their respective Boolean equations and in an active state in the case of the “OR-NOT”. As such, a modeler could make use of the link operator function bias to select the appropriate model parameterization which statistically guarantees an activity state profile that best matches the one supported by experimental evidence.

6.1 From topology to link operator Boolean models

In order to define Boolean models with the “AND-NOT” and “OR-NOT” link operator parameterization forms, we implemented the software *abmlog*, which stands for “All possible Boolean Models Link Operator Generator” ([Software and Data Availability](#)). Given a simple interaction (.sif) format file [7], representing a PKN with clearly defined, positive and negative causal interactions, the abmlog software outputs all combinatorially possible Boolean models where each link operator equation (deciding the state of a *link operator node*, i.e. one whose Boolean activity state is determined by both positive and negative regulators) will have either the “AND-NOT” or the “OR-NOT” function form. The models are saved in both the widely-used BoolNet (.bnet) [27] format and the gitsbe format [28], with the latter additionally including the attractors of the Boolean model, calculated via the BioLQM Java library [29]. A simple overview of the software is presented in Figure 2.

By default, abmlog generates all possible Boolean models with the two link operator parameterizations, the number of which depends on the number of link operator nodes. For example, if a network has 12 nodes with both activating and inhibitory regulators, then a total of $2^{12} = 4096$ Boolean models will be generated. In case the number of all possible Boolean models is very large or space restrictions do not allow the storage of that many models, the software can also be used to generate a random sample of link operator Boolean models from the total parameterization space. In summary, abmlog is a useful tool that can generate a large pool of Boolean models for subsequent analyses, each with a unique link operator parameterization.

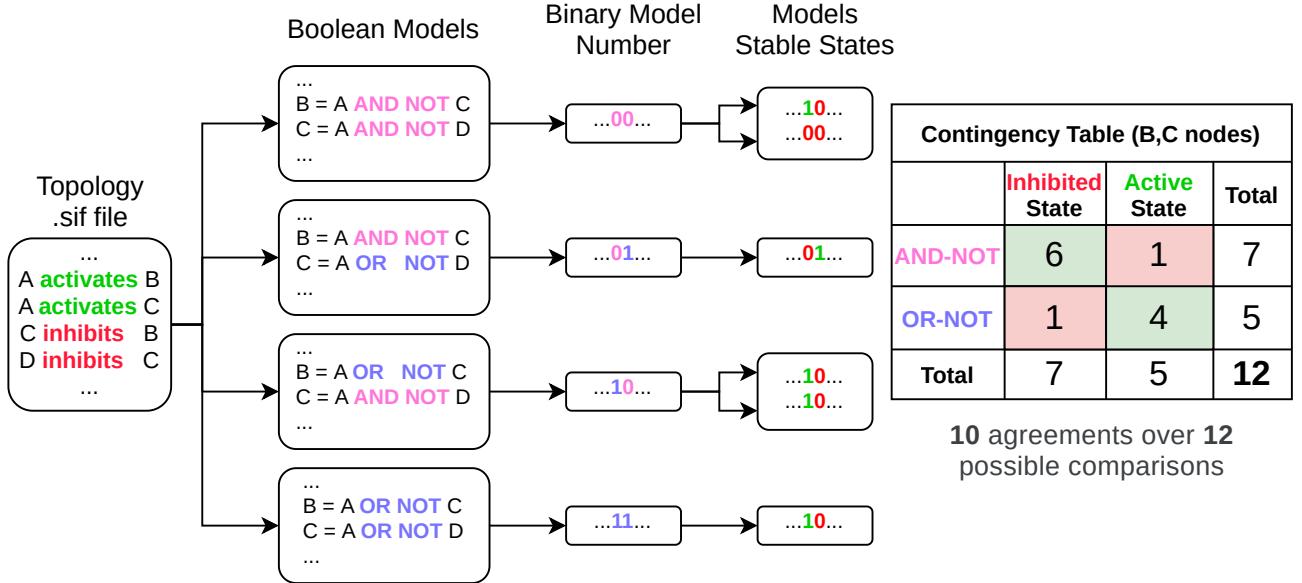


Figure 2: Data-flow overview diagram of the abmlog software and its related contingency table between output model parameterization and stable state activity. A simple interaction file is given as an input to produce a series of Boolean models where equations with both activating and inhibitory regulators have either the “AND-NOT” or the “OR-NOT” formulation. Two link operator equations give rise to a total of $2^2 = 4$ different Boolean models. Each unique parameterization can be represented by a single binary model number, where a “0” corresponds to an equation with the “AND-NOT” link operator and a “1” to an equation with the “OR-NOT”. This representation of parameterization can be directly compared to each of the models’ stable states, which enables the creation of a contingency table for the data pertaining to nodes B and C and the derivation of measures of agreement (see Section 6.2).

6.2 Measuring agreement between parameterization and stable state

In order to quantify the link operator function bias, we use measures of agreement between parameterization and stable state. The idea is that the more biased the link operator parameterization is, the higher the expected agreement will be between a target node’s link operator assignment and its corresponding stable state. For the rest of this work, we shall use two measures of agreement, namely the *percent agreement* and Cohen’s *kappa statistic* [30].

In more detail, using the Boolean model data generated by abmlog, we focus in two categorical variables related to a particular node of interest: its link operator parameterization (“AND-NOT”/“0” or “OR-NOT”/“1”) and its corresponding stable state activity (“inhibition” or “activation”), obtained via attractor analysis. We shall say that these two variables “agree” when a node whose target Boolean equation has the “AND-NOT” link operator (resp. “OR-NOT”) ends up with an inhibited (resp. active) state in the corresponding attractor. In the case of a Boolean model with multiple attractors, each of the stable states is used separately to measure the agreement between the two aforementioned variables, since the activity of a node might change between the different attractors, but its parameterization stays the same.

To define measures of agreement between the two proposed categorical variables, we visualize their interrelation using a contingency table. A total of four data comparison counts can be used to fill in the table’s cells: two where the parameterization and stable state match (i.e. node had the “AND-NOT” link operator form and an inhibited stable state or the “OR-NOT” form and an active state) and two where they differ (i.e. node had the “OR-NOT” form and its state was inhibited, or the “AND-NOT” form and an active state). The

percent agreement is then simply defined as the total number of matches divided by the total number of comparisons and is directly interpreted as the percentage of data that the two variables agree upon. In the example of Figure 2, the corresponding contingency table counts all the matches and mismatches between the link operator assignments for nodes B and C and their corresponding activity state (12 comparisons in total). Since there are only two mismatches, the percent agreement is equal to $10/12 = 0.83$, meaning that in 83% of the presented data, the link operator parameterization dictated function outcome. Naturally, a value of 0 is the absolute minimum score and indicates complete disagreement between the two variables while a perfect agreement score is equal to 1 or 100%.

A more robust statistic that we also apply in the Boolean model data is Cohen's kappa (κ) coefficient [30]. This statistic is used to measure the extent to which data collectors (raters) assign the same score to the same variable (inter-rater reliability) and takes into account the possibility of agreement occurring by chance. In our case, this can be conceived as one rater that assigns link operator parameterization ("AND-NOT" or "OR-NOT") and another that assigns stable state activity ("inhibition" or "activation"). Both variables are converted to a binary outcome (0 or 1), allowing the creation of a contingency table and subsequently the calculation of Cohen's formula for κ . The kappa statistic ranges from -1 to $+1$, where a value of 0 represents the amount of agreement that can be expected from random chance, and a value of 1 (resp. -1) indicates perfect agreement (resp. disagreement) between the raters. In the example contingency table of Figure 2, $\kappa = 0.657$, which is a considerable reduction in the level of congruence compared to the 0.83 percent agreement.

6.3 Truth Density bias in biological networks

6.3.1 Bias guides model parameterization in a cancer signaling network

We used abmlog on a cancer signaling network, consisting of 77 nodes and a total of 149 curated causal interactions that cover a variety of pathways linked to prosurvival and antisurvival cell signaling (e.g. cyclin expression and caspase activation). This PKN, named CASCADE (CAncer Signaling CAusality DatabasE), was successfully used to build a Boolean model able to predict anti-cancer drug combination effects in gastric cell lines [14]. We used the CASCADE version from the Flobak paper (version CASCADE 1.0), with some node naming changes for compatibility with the newest versions [31]. The number of nodes with both activating and inhibiting regulators in the CASCADE 1.0 topology is 23, while the rest of the nodes have regulators that belong to only one of the two regulatory categories. Thus, using abmlog, we generated all 2^{23} possible Boolean models with the "AND-NOT" and "OR-NOT" link operator parameterizations. The resulting stable state distribution across all produced models is presented in Figure 3A. For our subsequent analysis we will use only the 2802224 Boolean models that had exactly one stable state, as it makes the calculation of agreement between a node's assigned link operator and its corresponding activity state across all the selected models more straightforward.

The agreement results between link operator parameterization and stable state activity across all the selected CASCADE models are presented in Figures 3B (percent agreement, per node) and 3C (Cohen's κ , nodes with the same number of regulators are grouped together). The percent agreement results show a high variability across the link operator nodes and range from a minimum of 53% to a perfect agreement (100%). This suggests that for all nodes, across any selected CASCADE 1.0 Boolean model, there is a higher than random probability that the assignment of the "AND-NOT" (resp. "OR-NOT") link operator formula in the associated Boolean equations will result in the inhibition (resp. activation) of the target nodes. So, even though none of the nodes have more than 5 regulators, we already start seeing signs of the truth density bias in the link operator regulatory functions across a wide range of Boolean models.

When applying Cohen's κ to evaluate level of agreement, we chose a conservative threshold equal to 0.6, corresponding empirically to a substantial level of agreement [32, 33]. We found that 60% (14 out of 23) of

the nodes have a κ value below the specified threshold. Our conclusion is that biological networks with higher in-degree nodes (i.e. more than 7 – 10 regulators) are needed to properly assess if there is a truly high level of agreement between Boolean parameterization and function state outcome in the case of the link operator regulatory functions, providing thus conclusive proof of their bias (Section 6.3.2).

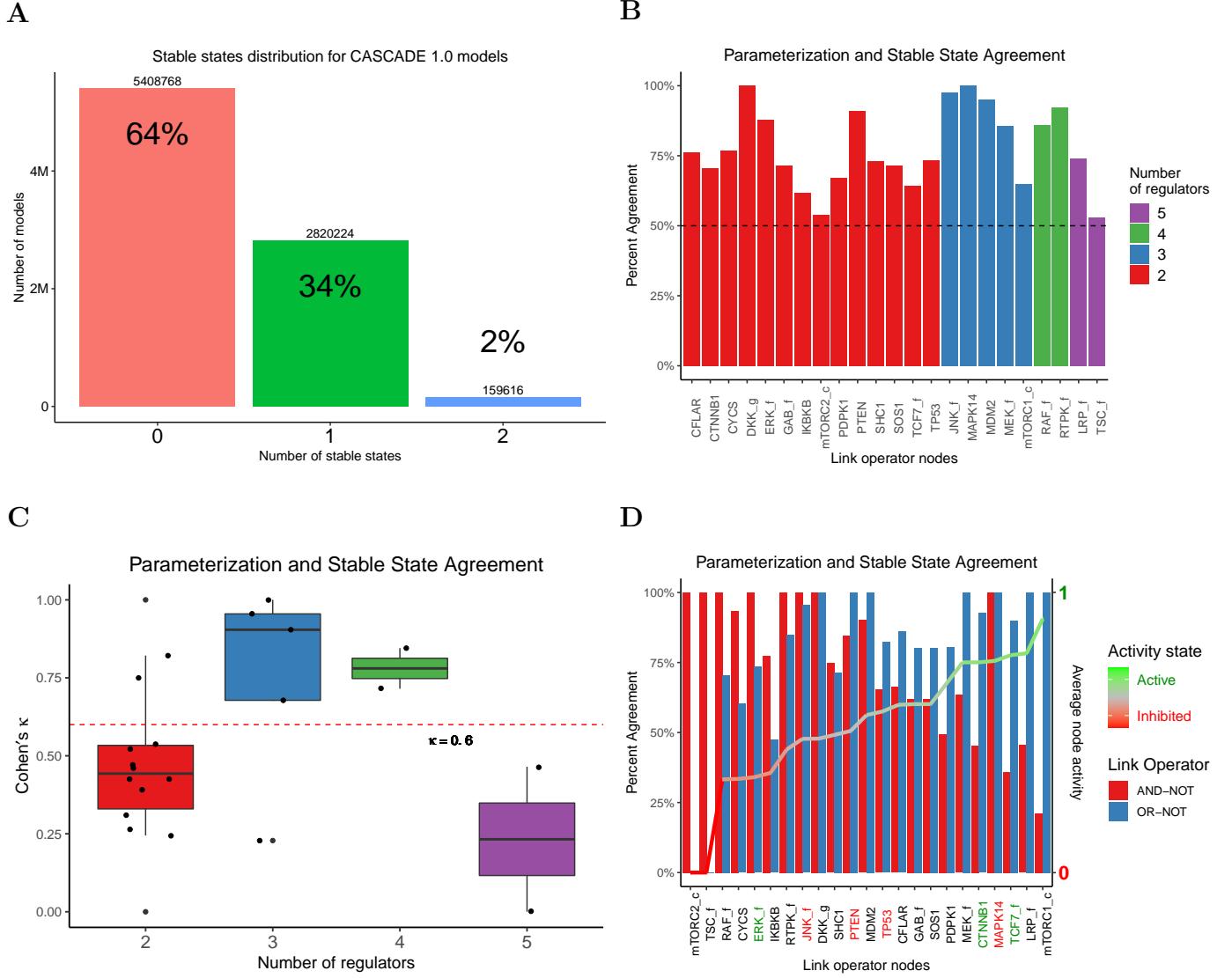


Figure 3: (A) Stable states distribution across all link operator parameterized Boolean models generated by the abmlog software using the CASCADE 1.0 signaling topology. (B) Percent agreement scores between parameterization and activity state across all single stable state CASCADE 1.0 models, for 23 nodes with both inhibiting and activating regulators. Nodes are sorted according to the total number of input regulators. (C) Same as (B), with the difference that the link operator nodes are now grouped into categories based on the total number of input regulators and Cohen's κ is used as an agreement statistic. (D) Same as (B), with the agreement now calculated as the proportion of matches between a node's link operator and its activity state, in the models that had the specific parameterization. The link operator nodes are sorted according to the average activity state across the considered CASCADE models and the colored node labels indicate literature curated activity profiles from Flobak et al. [14].

Regardless of the presence of bias or not, the agreement results can be used to show how experimental data and topological regulatory knowledge (e.g. the activator-to-inhibitor ratio) can be coupled with the truth density metric to guide the choice of regulatory functions. In one example scenario, a modeler asks what the most probable link operator parameterization is among the “AND-NOT” and “OR-NOT” forms that matches available experimental evidence. We used a literature curated activity profile derived for the AGS cell line from [14], to annotate 7 of the link operator nodes in Figure 3D according to their experimentally validated state (activation or inhibition). To clearly identify which of the two parameterizations best fits the observed data, for each node we split the CASCADE models in two model pools, representing the “AND-NOT” and “OR-NOT” node parameterizations, and calculated the proportion of models within each pool whose link operator matched the expected state outcome. For example, in the contingency table of Figure 2, the equivalent calculation would be to divide the number of matches in each row with the corresponding row total sum, resulting in $6/7 = 85.7\%$ of the “AND-NOT” Boolean equations with an inhibited stable state and $4/5 = 80\%$ of the “OR-NOT” equations with an active target node. Moreover, the link operator nodes of Figure 3D are sorted in increasing order by their average stable state activity in the considered CASCADE 1.0 Boolean models. It is evident that nodes with higher average activity in the stable state have a higher agreement with the “OR-NOT” parameterization whereas nodes with lower average activity, a higher agreement with the “AND-NOT” parameterization (0.85 and -0.74 Pearson correlation coefficients with $p_{corr}^{\text{OR-NOT}} = 2.6 \times 10^{-7}$ and $p_{corr}^{\text{AND-NOT}} = 5 \times 10^{-5}$ respectively, see [Software and Data Availability](#)).

More specifically, we observe that for all experimentally validated nodes, a modeler could a priori set the link operator to the appropriate form and get a stable state activation profile that matches the observations (“AND-NOT” to match an inhibition node profile or “OR-NOT” for an activation profile) with a higher probability than if he was randomly choosing one of the two. For example, the data shows that 90% of the models with an “OR-NOT” Boolean equation for the target family node TCF7_f, had the node as active in their respective stable state. The same is observed for the CTNNB1 (92%) and ERK_f active nodes (74%), as well as for the TP53 (65%) and PTEN (85%) inhibited nodes with the choice of the “AND-NOT” parameterization. Additionally, all the aforementioned nodes have two regulators (one activator and one inhibitor) and using the respective truth density formulas (Eq. 6 and 7) with $n = 2$ and $m = k = 1$, we have that $TD_{\text{AND-NOT}} = 0.25$ (closer to 0 or inhibition) and $TD_{\text{OR-NOT}} = 0.75$ (closer to 1 or activation), as was also shown in Figure 1A. As such, the nodes observed output matches the statistically expected binary outcomes, showing that even with a low number of regulators, the BRF bias can be used to guide function choice.

In another scenario, a modeler knows that a particular node has a skewed activator-to-inhibitor ratio and wants to exploit such knowledge to make the node conform to a particular activity state of his choice. A nice example from our data is the family node LRP_f, with four activators and one inhibitor. Using the truth density formulas for the two link operator parameterizations (Eq. 6 and 7) with $n = 5$, $m = 4$ and $k = 1$, we have that $TD_{\text{AND-NOT}} = 0.47$ and $TD_{\text{OR-NOT}} = 0.97$. So, if the modeler wants to have an active LRP_f in the stable state, the “OR-NOT” parameterization should be preferred since the “AND-NOT” has an approximate 50% probability for this to happen from a statistical point of view. These truth density values also match the results from Figure 3D, since only half of the models that use the “AND-NOT” parameterization end up with an inhibited LRP_f in the stable state while all of them have an active LRP_f (100% agreement) in the case where the “OR-NOT” form is used. Also, the average activity of LRP_f across all models is one of the highest in the data, suggesting that imbalanced activator-to-inhibitor ratios could be a direct proxy for predicting regulation outcome. In a similar situation, but at the other range of the activity spectrum, we have the TSC_f family node with one activator and four inhibitors. The truth density values (now using $n = 5$, $m = 1$ and $k = 4$) are $TD_{\text{AND-NOT}} = 0.03$ and $TD_{\text{OR-NOT}} = 0.53$ respectively. Therefore, the “AND-NOT” parameterization guarantees the inhibition of the TSC_f node (data shows 100% agreement) and it should be a modeler’s first choice if that is the desired outcome. On the other hand, if the activation of TSC_f was a modeler’s preference, then the choice of the “OR-NOT” form would be the most

statistically appropriate according to the truth density metric. We observe though that there was no model having `TSC_f` inhibited in the stable stable, indicating that the complex dynamics of the cancer network can also play a significant role in the function outcome. In general, we note that the particular configuration of activating and inhibiting regulators of a target in a specific model instance, can influence the dynamics attributable to the parameterization, causing several results from our analysis to differ from the expected behavior of the Boolean functions studied.

6.3.2 Hub node bias in random scale-free networks

In the previous section we showed that the truth density bias can be used to predict regulatory function outcome in a specific cancer signaling network, but the question still remains open for general biological networks. Also, we found evidence suggesting that Boolean dynamics also plays a significant role in deciding each node's state in the attractors and in some cases activity state results may contradict what is expected from the use and asymptotic interpretation of the truth density formulas. Therefore, we now proceed to investigate if networks with higher in-degree nodes (i.e. more input regulators) have stable states that can be unquestionably decided a priori by the truth density metric, using the respective *TD* formulas for the “AND-NOT” and “OR-NOT” link operator parameterizations.

We study the specific class of scale-free networks [34], based on the hypothesis that most biological networks exhibit that property, i.e. their node degree distribution follows asymptotically a power law $P(k) \sim k^{-\gamma}$, with k the number of regulators and γ the scale-free exponent. We note that the CASCADE 1.0 model also exhibits the scale-free property (see [Software and Data Availability](#)) and there has been evidence in the literature both in favor and against this hypothesis. In particular, earlier studies showed that many complex networks (including metabolic ones) are approximately scale-free [35, 36, 37, 38], whereas more recent efforts demonstrated that not all cellular biological networks may share that property [39], but those that do, exhibit the strongest level of evidence of scale-free structure [40]. Consequently, we shall use scale-free topologies as acceptable substitutes of real biological networks in our analysis.

The methodology is as follows: we start by generating scale-free topology files with a total of 50 nodes each and a maximum in-degree $k_{max} = 50$ [27]. For each network, the number of input regulators per node is drawn from a Riemann Zeta distribution with parameter γ [41]. The choice of regulators for each network node, as well as the type of regulation (positive or negative), is uniformly random. The Zeta distribution allows the creation of in-degree values that far exceed the average connectivity in a network, giving rise to the highest-degree nodes (often called “hubs”), which are the most defining characteristic of the scale-free networks. The value of the scale-free exponent influences the number of hubs and their in-degree distribution. More specifically, we created scale-free networks with $\gamma = 2$ and $\gamma = 2.5$, since most of the studied networks have an exponent between 2 and 3 [41, 42]. Comparing the networks built with the above methodology, we found that those with $\gamma = 2$ have more nodes with both activating and inhibiting regulators and higher degree hubs than networks with $\gamma = 2.5$ (Figures 4A and 4B). These two characteristics suggest that the scale-free networks with $\gamma = 2$ are better suited for use with the abmlog software, since the larger the number of link operator nodes, the more Boolean models can be generated and thus more data comparisons can be made between node parameterization and stable state activity. Additionally, the presence of higher degree hubs is the perfect testbed for the link operator function bias, which manifests especially for nodes with more than 7 – 10 regulators, as we found from our earlier truth density asymptotics analysis (Figure 1A).

Our methodology proceeds with using each of the scale-free topologies with $\gamma = 2$ as input to the abmlog software, and generating ensembles of Boolean models parameterized with every possible mix of the “AND-NOT” and “OR-NOT” regulatory functions along with the calculation of their stable states (as demonstrated in Figure 2). The produced Boolean models had zero, one, or more stable states. Interestingly, we observed that around half of the tested scale-free topologies generated Boolean models with no stable states, no matter which combination of link operators was used to define the model parameterization. Therefore, the randomly

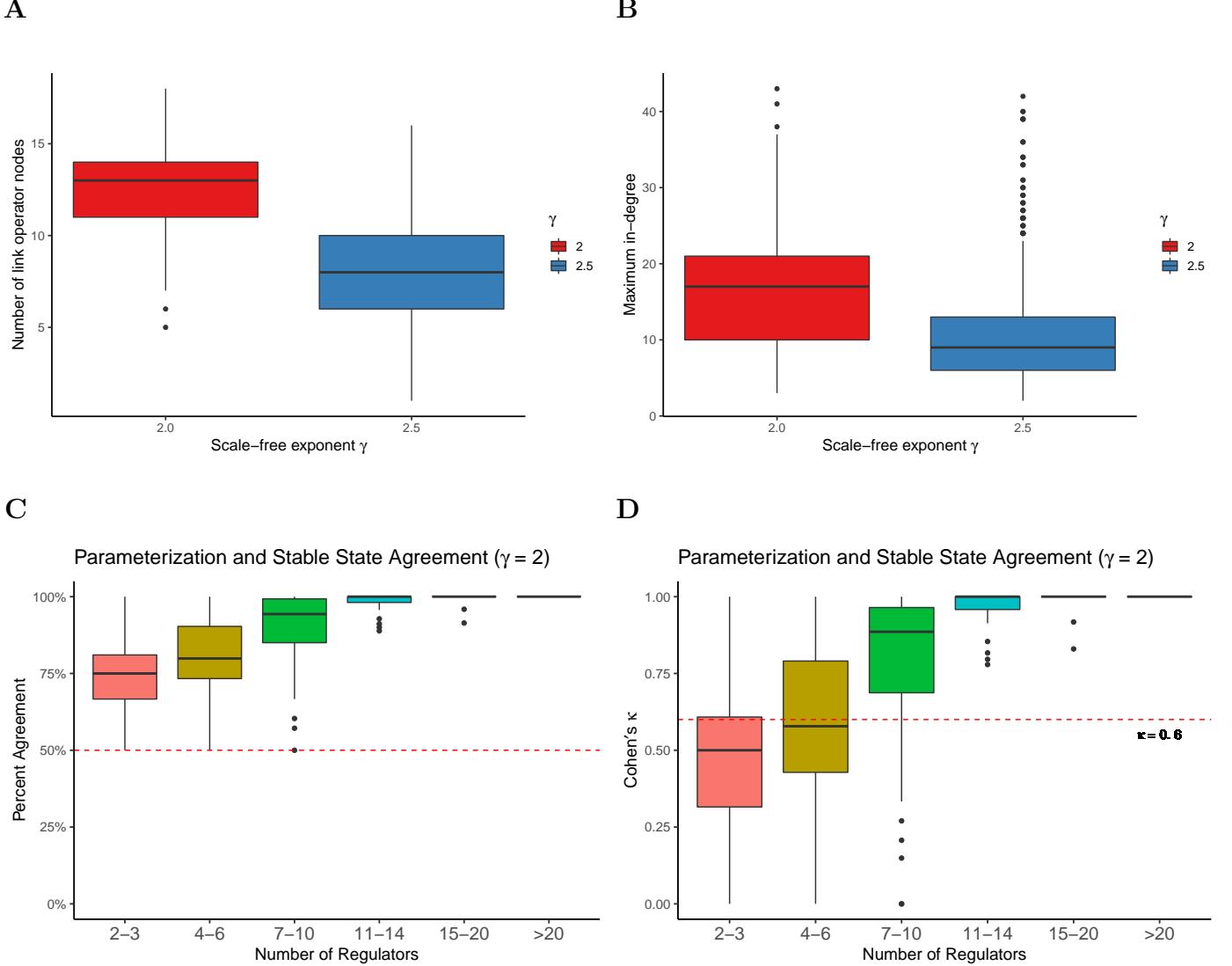


Figure 4: (A)-(B) Network statistics for scale-free topologies with different degree exponents. Every network tested has 50 nodes and a maximum in-degree $k_{max} = 50$. A total of 100 topologies for $\gamma = 2$ and 1000 topologies for $\gamma = 2.5$ are compared. Networks with $\gamma = 2$ have a higher median number of nodes with both activating and inhibiting regulators and higher degree hubs. (C)-(D) Agreement statistics between link operator parameterization and stable state activity. The data is taken from Boolean models generated with the abmlog software, using scale-free topologies with exponent $\gamma = 2$. A total of 757 link operator nodes were compared across multiple link operator parameterization configurations with their corresponding stable states. Nodes are grouped in buckets, where each bucket indicates a different range of input regulators. Both the percent agreement and Cohen's κ show considerable congruence between link operator assignment (“AND-NOT” or “OR-NOT”) and resulting stable state (inhibition or activation respectively) for nodes with more than 10 input regulators.

assigned regulators, regulatory effects, and Zeta distribution in-degree values, may result in networks which do not have stable phenotypes, suggesting that alternative parameterizations might be more suitable in modeling scenarios which specifically examine stable dynamics. Nonetheless, we discarded the models with no stable states and used the rest that had single or multiple attractors in our analysis. Then, for each model

node with both activating and inhibiting regulators, we compared its assigned link operator with the activity state value in the corresponding stable state(s), across all the link operator parameterization spectrum that yielded models with stable phenotypes. The agreement results between parameterization and stable state activity are presented in Figure 4C for the percent agreement and in Figure 4D for Cohen’s kappa statistic.

We observe that both presented statistics show a large variation of agreement for nodes with less than 10 regulators and an increasing agreement with more regulators. This agreement manifests in link operator nodes parameterized with the “AND-NOT” or “OR-NOT” Boolean functions, while at the same time exhibiting inhibited or active states respectively in the associated model attractors. Therefore, we conclude that the considered standardized Boolean regulatory functions are biased and their outcomes can be determined a priori from the choice of the corresponding link operator parameterization, especially for nodes with more than 7 – 10 regulators.

7 Discussion

The specification of mathematical rules that describe the behavior of biological systems is one of the core aspects of computational modeling. It is therefore of considerable value to have a list of metrics that can be used to compare different model parameterizations and make an informed decision with regard to the selection of an appropriate regulatory function that better matches the expected behavior in a specific modeling application.

We specifically discussed two characterizations that can assist modelers in comparing various regulatory functions and select the most plausible ones with regard to the causal interaction-based knowledge at hand. Expressing Boolean functions in DNF makes biological interpretation concrete by explicitly specifying the conditions (presence or absence of the positive and negative regulators, respectively) that make a target active. Expressing the functions in CDNF allows to easily check for compliance with the underlying regulatory topology and subsequently, the rejection of functions that violate such consistency. The difference between these two characterizations lies in the fact that the consistency terminology stems from the mathematical world, while biological interpretability is tightly connected to the world of language semantics and thus closer to the modeler’s point of view. Finally, truth density is an informative measure which can be used to verify if the function parameterization dictates biased Boolean outcomes. It can also be used as a test metric to understand how a function behaves when the number of regulators increases or the balance between the number of activators and inhibitors changes.

Using the truth density metric, we showed the presence of link operator function bias in the hubs of randomly constructed scale-free networks. A potential application of this finding could be to dramatically decrease the time needed to train Boolean models to fit observations via various optimization methods, by pre-assigning the parameterization of link operator nodes with sufficiently many regulators. The pruning of the searchable parameterization space, guided by the truth density metric, can result in more efficient automated methods and can enable the training of larger models against data from numerous resources (e.g. large cell line panels). The hub node bias has also interesting links to the presence of order in biological networks [43]. The dynamics of a Boolean network can exhibit ordered or chaotic behavior. Ordered dynamics is characterized by the presence of less stable states and limit cycle attractors with smaller mean length (number of states in a complex attractor) and transition times (number of steps needed to reach an attractor starting out from an arbitrary configuration) [41]. It is also known that the truth density (probability of target expression) as well as the degree exponent γ (related to network connectivity) can modulate the dynamic transition between the ordered and chaotic phases. Moreover, it has been shown that above the critical value of $\gamma_c \sim 2.47$, ordered behavior in the form of stable state dynamics manifests independently of the truth density, whereas for values closer to $\gamma = 2$, order coincides with the presence of high biased nodes (see Fig. 4 in [41]). Our work confirms this phenomenon, since the use of the link operator parameterization guarantees the presence of biased hubs,

which enable the scale-free networks to exhibit stability and homogeneity in terms of regulatory output, and thus stay in the ordered dynamic regime.

Searching for other function metrics that are applicable to logical modeling, the *sensitivity* of a Boolean function is one of the most relevant [44]. As its name suggests, it measures how sensitive the output of the function is to small changes of its inputs. Sensitivity is tightly linked to the truth density metric, since a highly homogeneous Boolean function (i.e. a biased one), is unlikely to change its value between similar regulatory input configurations and so, its sensitivity is relatively low. To compute the average sensitivity value for an arbitrary Boolean function we need to sum over all the *influences* of the input variables, which essentially represent a way to measure individual variable importance. In the context of regulatory functions, a regulator's influence is defined as the probability that a random toggle on its activity (from active to inactive and vice-versa) will change the value of the Boolean function [45]. Therefore, by calculating the influence of every regulator, the modeler can gain knowledge of which ones are more important and control the respective function's outcome. This transition of perspective from the function level to the regulator level might be advantageous in cases where the modeler's intention is to compare different parameterizations and choose the one for which a particular regulator is labeled as significantly more important than the others, based on the available biological knowledge.

Lastly, an important addition to a universal list of Boolean function metrics for modeling purposes, is the notion of function *complexity*. A recent definition is given by Gherardi et al. [25], where the authors defined it as the number of terms in the shortest possible DNF expression of a given Boolean function, divided by the total number of rows in the corresponding truth table. We presented this information in the last column of Table 2, where the BRFs are sorted from lower to higher complexity (note that the CDNF has the minimum number of terms for every BRF included in the table). One useful observation is that the standardized “AND-NOT” formula [9] is the function with the lowest complexity that is also consistent and thus biologically plausible - all properties that make it a good choice from the modeler’s perspective. Assessing the complexity of the studied regulatory functions using the derived formulas for the minimum number of CDNF terms for any number of activators m and inhibitors k (see last column of Table 2), we comment on the fact that all BRFs have very low complexity since $\mathcal{O}(m \times k) \ll 2^{m+k}$, i.e. the number of function terms does not grow as fast as the number of rows in the corresponding truth table. Same observation has been shown to be true in manually-tuned, experimentally-validated Boolean functions [25], providing us with another confirmation that the consistent functions from Table 2 are good candidates for logic-based modeling approaches.

8 Future work

In this work we make an attempt to address the logical rule specification problem, which can be simply stated as: “Many functions may fit the available observations, which one is the most proper to use?” Of course what is “proper” can be fairly subjective, but the main point is that a careful consideration of the underlying application context (i.e. what output do I expect in a specific scenario of interest) along with a list of metrics that explicate a Boolean function’s behavior and semantics, provides the user with the appropriate framework to decide on the function parameterization that sets the basis for further model analysis and simulation. In that regard, interesting directions for further research include the application of the metrics presented in this work in different published biological models, and the subsequent comparison of different regulatory functions within this framework. Such meta-analyses could potentially indicate regulatory functions that achieve a higher degree of fitness with the observed data or general properties that are common in all Boolean functions used to model biological systems.

An interesting study for example would be to analyze Boolean functions from published biological models that have extreme activator-to-inhibitor ratios. If such imbalanced ratios also result in proportionally skewed Boolean outcomes (i.e. with more activators, the truth density is closer to 1 and the reverse with more inhibitors), suggesting that target outcome follows the majority regulatory groups, then the use of threshold

functions could be a more proper parameterization alternative, as was shown in Figure 1B. Of course, we note that each individual case must be examined with care, since there might be high influence nodes, whose activity defines the target’s output even in the presence of a much larger regulatory group with opposite effects. For example, CASP3 is a biological entity that, when activated, will almost certainly result in the cell’s death even in the presence of a majority of proliferation-positive regulators at any given time. Subsequently, a more appropriate choice based on the results of this study can be made, either by choosing between the biased functions, which demonstrate a more balanced behavior for such extreme activator-to-inhibitor ratios (e.g. using the “Pairs” or the “AND-NOT” functions which are balanced vs using the “OR-NOT” which would make the target activated most of the time, see Scenario 2) or by using refined threshold functions, in which each regulator’s weight will differ in order to match the influence that it has on the target.

There have been only a handful examples of published logical models [46, 47] and research papers [21, 48, 49, 50, 51, 52, 53, 54] that use the threshold modeling framework in biological systems. This is partly due to the lack of tools that make threshold functions accessible to the average user, and the availability of such software in open-source environments such as the CoLoMoTo Interactive Notebook [55]. We believe that the existence of such novel software will enable the construction and configuration of generic Boolean threshold models and provide users of the logical-modeling community and beyond with the necessary toolbox to further study these models. This will enable applications that depend on the dynamical analysis of Boolean threshold models (identification of attractors, reachability properties, formal verification and control) and the use of optimization methods to calibrate the threshold function parameters to best fit the available experimental data, as is done currently with analytical logic-based functions [12].

Software and Data Availability

The *abmlog* software that was used to generate Boolean models with the “AND-NOT” and “OR-NOT” Boolean regulatory functions is available at <https://github.com/druglogics/abmlog> under the MIT License. We used the version 1.6.0 for this analysis, which is also offered as a standalone package at <https://github.com/druglogics/abmlog/packages>.

An extended analysis accompanying the results of this paper is available at <https://druglogics.github.io/brf-bias>. It includes links to the produced model datasets and scripts to reproduce the results and figures of this paper. In particular, the correlation analysis between average node state in the CASCADE 1.0 models and percent agreement per each link operator is available at <https://druglogics.github.io/brf-bias/cascade-1-0-data-analysis.html#node-state-and-percent-agreement-correlation>. The degree distribution of the CASCADE 1.0 topology and other network statistics are examined in <https://druglogics.github.io/brf-bias/cascade-1-0-data-analysis.html#network-properties>.

Funding

This work was supported by ERACoSysMed Call 1 project COLOSYS (JZ, MK), project UIDB/50021/2020 from Fundação para a Ciência e a Tecnologia - INESC-ID multi-annual funding (PM), the Norwegian University of Science and Technology’s Strategic Research Area ‘NTNU Health’ and The Joint Research Committee between St. Olavs hospital and the Faculty of Medicine and Health Sciences, NTNU - FFU (ÅF).

Conflict of Interest: none declared.

Acknowledgments

The authors acknowledge Dr. Vasundra Touré for her contribution in the improvement of Figure 2 and its caption.

A Truth Density formula proofs

For all the following propositions, we consider f to be a Boolean regulatory function $f_{BRF} : \{0, 1\}^n \rightarrow \{0, 1\}$, with a total of n input regulators separated to two distinct sets, the set of $m \geq 1$ activators $x = \{x_i\}_{i=1}^m$ and the set of $k \geq 1$ inhibitors $y = \{y_j\}_{j=1}^k$, such that $n = m + k$.

Proposition 1 (“AND-NOT” Truth Density). *The truth density of the “AND-NOT” link operator function $f_{AND-NOT}(x, y) = (\bigvee_{i=1}^m x_i) \wedge \neg(\bigvee_{j=1}^k y_j)$, with $m \geq 1$ activators and $k \geq 1$ inhibitors, is given by the formula:*

$$TD_{AND-NOT} = \frac{2^m - 1}{2^n} = \frac{1}{2^k} - \frac{1}{2^n} \quad (6)$$

Proof. Using the distributive property and De Morgan’s law we can express $f_{AND-NOT}$ (Eq. 1) in the equivalent DNF:

$$\begin{aligned} f_{AND-NOT}(x, y) &= \left(\bigvee_{i=1}^m x_i \right) \wedge \neg \left(\bigvee_{j=1}^k y_j \right) \\ &= \bigvee_{i=1}^m \left(x_i \wedge \neg \left(\bigvee_{j=1}^k y_j \right) \right) \\ &= \bigvee_{i=1}^m \left(x_i \wedge \bigwedge_{j=1}^k \neg y_j \right) \\ &= \bigvee_{i=1}^m (x_i \wedge \neg y_1 \wedge \dots \wedge \neg y_k) \end{aligned}$$

To calculate $TD_{AND-NOT}$, we need to find the number of rows in $f_{AND-NOT}$ ’s truth table that result in a *True* output result and divide that by the total number of rows, which is 2^n (n input regulators).

Note that $f_{AND-NOT}$, written in its equivalent DNF, has exactly m terms. Each term has a unique *True/False* assignment of regulators that makes it *True*. This happens when the activator of the term is *True* and all of the inhibitors *False*. Since the condition for the inhibitors is the same regardless of the term we are examining and f is expressed in DNF, the *True* outcomes of the function f are defined by all logical assignment combinations of the m activators that have at least one of them being *True* and all inhibitors assigned as *False*. There are a total of 2^m possible *True/False* logical assignments of the m activators (from all *False* to all *True*) and $f_{AND-NOT}$ becomes *True* on all except one of them (i.e. when all activators are *False*), with the corresponding $2^m - 1$ truth table rows having all inhibitors assigned as *False*. Therefore, $TD_{AND-NOT} = (2^m - 1)/2^n$. \square

Proposition 2 (“OR-NOT” Truth Density). *The truth density of the “OR-NOT” link operator function $f_{OR-NOT}(x, y) = (\bigvee_{i=1}^m x_i) \vee \neg(\bigvee_{j=1}^k y_j)$, with $m \geq 1$ activators and $k \geq 1$ inhibitors, is given by the formula:*

$$TD_{OR-NOT} = \frac{2^n - (2^k - 1)}{2^n} = 1 - \frac{1}{2^m} + \frac{1}{2^n} \quad (7)$$

Proof. Using De Morgan's law we can express f_{OR-NOT} (Eq. 2) in the equivalent DNF:

$$\begin{aligned} f_{OR-NOT}(x, y) &= \left(\bigvee_{i=1}^m x_i \right) \vee \neg \left(\bigvee_{j=1}^k y_j \right) \\ &= \left(\bigvee_{i=1}^m x_i \right) \vee \left(\bigwedge_{j=1}^k \neg y_j \right) \\ &= x_1 \vee x_2 \vee \dots \vee x_m \vee (\neg y_1 \wedge \dots \wedge \neg y_k) \end{aligned}$$

To calculate TD_{OR-NOT} , we find the number of rows of f_{OR-NOT} 's truth table that result in a *False* output (R_{false}), subtract that number from the total number of rows (2^n) to get the rows that result in f being *True*, and then divide by the total number of rows. As such, $TD_{OR-NOT} = (2^n - R_{false})/2^n$.

Note that f_{OR-NOT} , expressed in it's equivalent DNF, has exactly $m + 1$ terms. To make f_{OR-NOT} *False*, we assign the m activators as *False* and then we investigate which logical assignments of the inhibitors $\{y_j\}_{j=1}^k$ make the last DNF term also *False*. Out of all the possible 2^k *True/False* logical assignments of the k inhibitors (ranging from all *False* to all *True*) there is only one that does not make the last term of f_{OR-NOT} *False*, which happens specifically when all k inhibitors are *False*. Therefore, $R_{false} = 2^k - 1$ and $TD_{OR-NOT} = (2^n - (2^k - 1))/2^n$. \square

Proposition 3 (“Pairs” Truth Density). *The truth density of the “Pairs” link operator function $f_{Pairs}(x, y) = \bigvee_{\forall(i,j)}^{m,k} (x_i \wedge \neg y_j)$, with $m \geq 1$ activators and $k \geq 1$ inhibitors, is given by the formula:*

$$TD_{Pairs} = \frac{(2^m - 1)(2^k - 1)}{2^n} \quad (8)$$

Proof. Using the distributive property we can express f_{Pairs} (Eq. 3) in its equivalent conjunction normal form (CNF), where two separate clauses are connected with AND's (\wedge) and inside the clauses the literals are connected with OR's (\vee):

$$f_{Pairs}(x, y) = \bigvee_{\forall(i,j)}^{m,k} (x_i \wedge \neg y_j) = \left(\bigvee_{i=1}^m x_i \right) \wedge \left(\bigvee_{j=1}^k \neg y_j \right) \quad (9)$$

To calculate TD_{Pairs} , based on its given CNF, we find the number of rows in its truth table that have at least one *True* activator (R_{act}) and subtract from these the rows in which all inhibitors are *True* (R_{inh}). Therefore, only the rows that have at least one *True* activator and at least one *False* inhibitor will be left, corresponding to the biological interpretation of f_{Pairs} . As such, $TD_{Pairs} = (R_{act} - R_{inh})/2^n$.

R_{act} can be found by subtracting from the total number of rows (2^n), the rows that have all activators as *False*. The number of these rows depends on the number of inhibitors, since for each one of the total possible 2^k *True/False* logical assignments of the k inhibitors (ranging from all *False* to all *True*), there will be a row in the truth table with all activators as *False*. Therefore, $R_{act} = 2^n - 2^k = 2^{m+k} - 2^k = 2^k(2^m - 1)$.

R_{inh} depends on the number of activators, since for each one of the total possible 2^m *True/False* logical assignments of the m activators (ranging from all *False* to all *True*), there will be a row in the truth table with all inhibitors as *True*. Note that we have to exclude one row from this result, which is exactly the row that has all activators as *False* since it's not included in the R_{act} rows. Therefore, $R_{inh} = 2^m - 1$ and $TD_{Pairs} = (R_{act} - R_{inh})/2^n = (2^k(2^m - 1) - (2^m - 1))/2^n$. \square

Proposition 4 (Threshold functions Truth Density). *The truth density of the Boolean threshold functions “Act-win” (Eq. 4) and “Inh-win” (Eq. 5), with $m \geq 1$ activators and $k \geq 1$ inhibitors, is given by the formula:*

$$TD_{thres} = \frac{\sum_{i=1}^m \left[\binom{m}{i} \times \sum_{j=0}^{\min(u,k)} \binom{k}{j} \right]}{2^n} \quad (10)$$

where $u = i$ or $i - 1$, depending on the use of the “Act-win” or “Inh-win” function respectively.

Proof. The truth density formula can be easily derived from the observation that we need to count the number of rows in the respective truth table that have more *True* activators than *True* inhibitors. In the case of the “Act-win” function, we also need to add the rows that have an equal number of *True* regulators in each respective category.

Firstly, we count all the subset input configurations that have up to m activators assigned to *True*. These include the partial *True/False* logical assignments that have either a single *True* activator, a pair of *True* activators, a triplet, etc. This is exactly the term $\sum_{i=1}^m \binom{m}{i}$. Note that each of these activator input configurations is multiplied by a factor of 2^k in the truth table to make *complete* rows, i.e. rows where the activators logical assignments stay unchanged and the inhibitor values range from all *False* to all *True*. Therefore, we need to specify exactly which inhibitor logical assignments are appropriate for each activator subset input configuration. To do that, we multiply the size of each activator subset $\binom{m}{i}$ with the number of configurations that have less *True* inhibitors, i.e. $\sum_{j=0}^{i-1} \binom{k}{j}$.

Let’s consider an example with $m, k > 2$ and set $i = 2$. We find that the number of subsets with 2 *True* activators is $\binom{m}{2}$. Next, we multiply by the number of configurations that have one or no *True* inhibitors, i.e. $\sum_{j=0}^1 \binom{k}{j}$. This results in the number of rows of interest for the “Inh-win” function, i.e. the rows where there are exactly 2 activators assigned to *True* and less than 2 *True* inhibitors. For “Act-win”, we have to multiply up to the *True* inhibitor pairs, i.e. $\sum_{j=0}^2 \binom{k}{j}$. In summation, we count the configurations that have exactly i out of m activators assigned to *True*, and for each one, we multiply by the number of cases that have 0 up to i inhibitors assigned to *True* to find the respective rows, i.e. $\binom{m}{i} \times \sum_{j=0}^i \binom{k}{j}$. Repeating this calculation for every possible subset of i activators (from 1 up to all m of them), and summing the rows up, will result in the numerator of the TD_{thres} formula for the “Act-win” function.

Lastly, note that the *largest* inhibitor configuration subset size that we consider, is the minimum value between the current activator subset size ($u = i$ or $i - 1$, depending on which threshold function we use) and the total number of inhibitors k . Therefore, we take into account the case where the number of inhibitors is less than the activator subset size, i.e. $k < u$. This explains the term $\min(u, k)$ in the truth density formula and concludes the proof. \square

B Truth Density asymptotic behavior

We study the asymptotic behavior of the four truth density formulas (Appendix A) for a large number of regulators ($n \rightarrow \infty$). Note that for the calculations involving the two threshold functions, we will only use the truth density formula corresponding to the “Act-win” function (Eq. 10, with $u = i$), since both functions have similar formulas and therefore, their limiting behavior is analogous. The asymptotics results for each regulatory function are as follows:

1. The “AND-NOT” function truth density (Eq. 6) depends only on the number of inhibitors k :

$$TD_{AND-NOT} = \frac{1}{2^k} - \frac{1}{2^n} \sim \frac{1}{2^k} \quad (11)$$

For large k , it is biased towards 0:

$$TD_{AND-NOT} = \frac{1}{2^k} \xrightarrow{k \rightarrow \infty} 0$$

2. The “OR-NOT” function truth density (Eq. 7) depends only on the number of activators m :

$$TD_{OR-NOT} = 1 - \frac{1}{2^m} + \frac{1}{2^n} \sim 1 - \frac{1}{2^m} \quad (12)$$

For large m , it is biased towards 1:

$$TD_{OR-NOT} = 1 - \frac{1}{2^m} \xrightarrow{m \rightarrow \infty} 1$$

3. The “Pairs” function truth density (Eq. 8) depends on both activators and inhibitors:

$$TD_{Pairs} = \frac{(2^m - 1)(2^k - 1)}{2^n} = \frac{2^n - 2^m - 2^k + 1}{2^n} = 1 - \frac{2^m + 2^k}{2^n} + \frac{1}{2^n} \sim 1 - \frac{1}{2^k} - \frac{1}{2^m} \quad (13)$$

4. The threshold functions truth density (Eq. 10) depends on both m and k variables and does not have a single fixed limit for $n \rightarrow \infty$.

We now focus on the effect of the ratio ($m : k$) between number of activators and inhibitors on the asymptotic truth density values for $n \rightarrow \infty$. We consider the following three scenarios for each of the Boolean functions:

Scenario 1 A 1 : 1 activator-to-inhibitor ratio, where approximately half of the regulators are activators and half are inhibitors, i.e. $m \approx k \approx n/2$ (consider n is even without loss of generality).

1. The “AND-NOT” function truth density is biased towards 0:

$$(Eq. 11) \Rightarrow TD_{AND-NOT} \sim \frac{1}{2^{n/2}} \xrightarrow{n \rightarrow \infty} 0$$

2. The “OR-NOT” function truth density is biased towards 1:

$$(Eq. 12) \Rightarrow TD_{OR-NOT} \sim 1 - \frac{1}{2^{n/2}} \xrightarrow{n \rightarrow \infty} 1$$

3. The “Pairs” function truth density is biased towards 1:

$$(Eq. 13) \Rightarrow TD_{Pairs} \sim 1 - \frac{1}{2^{n/2}} - \frac{1}{2^{n/2}} \xrightarrow{n \rightarrow \infty} 1$$

4. The threshold functions truth density is balanced, meaning its limit asymptotically approaches 1/2.

Proof. We first rewrite the truth density formula substituting $m = k = n/2$:

$$(Eq. 10) \Rightarrow TD_{thres} = \frac{\sum_{i=1}^{n/2} \left[\binom{n/2}{i} \times \sum_{j=0}^{\min(i, n/2)} \binom{n/2}{j} \right]}{2^n} = \frac{\sum_{i=1}^{n/2} \left[\binom{n/2}{i} \times \sum_{j=0}^i \binom{n/2}{j} \right]}{2^n} = \frac{N}{2^n}$$

Next we simplify N , by using the notation $z = n/2$ and \mathbf{x} as a meta-symbol for $\binom{z}{x}$. For example, $\binom{n/2}{1} = \binom{z}{1} = \mathbf{1}$. N is therefore expressed as:

$$N = \mathbf{1}(\mathbf{0} + \mathbf{1}) + \mathbf{2}(\mathbf{0} + \mathbf{1} + \mathbf{2}) + \dots + z(\mathbf{0} + \mathbf{1} \dots + z)$$

Using the symmetry of binomial coefficients: $\binom{z}{x} = \binom{z}{z-x} \sim \mathbf{x} = z - x$, we can re-write N as:

$$N = (z - \mathbf{1})[z + (z - \mathbf{1})] + (z - \mathbf{2})[z + (z - \mathbf{1}) + (z - \mathbf{2})] + \dots + \mathbf{0}[z + \dots + \mathbf{0}]$$

Adding the two expressions for N we have that:

$$2N = [\mathbf{0} + \mathbf{1} \dots + \mathbf{z}]^2 + \mathbf{1}^2 + \mathbf{2}^2 + \dots + (\mathbf{z} - \mathbf{1})^2 = 2^{2z} + \sum_{x=1}^{z-1} \mathbf{x}^2$$

Substituting back $\binom{z}{x} = \mathbf{x}$ and $i = x$ (change of index) in expression N , we have that the threshold functions truth density is written as:

$$TD_{thres} = \frac{N}{2^{2z}} = \frac{(1/2) \left[2^{2z} + \sum_{i=1}^{z-1} \binom{z}{i}^2 \right]}{2^{2z}}$$

As $n \rightarrow \infty$ (and hence $z \rightarrow \infty$), the term $\sum_{i=1}^{z-1} \binom{z}{i}^2$ does not grow as fast as 2^{2z} - it is smaller by a factor of $\sqrt{\pi z}$ (see answer to Problem 9.18 in [56]), and so it becomes negligible:

$$\lim_{z \rightarrow \infty} TD_{thres} = \lim_{z \rightarrow \infty} \frac{(1/2)2^{2z}}{2^{2z}} = \frac{1}{2}$$

□

Scenario 2 A high activator-to-inhibitor ratio ($n - 1 : 1$), where all regulators are activators except one inhibitor, i.e. $m = n - 1, k = 1$.

1. The “AND-NOT” function truth density is balanced:

$$(Eq. 11) \Rightarrow TD_{AND-NOT} \sim \frac{1}{2^1} = \frac{1}{2}$$

2. The “OR-NOT” function truth density is biased towards 1:

$$(Eq. 12) \Rightarrow TD_{OR-NOT} \sim 1 - \frac{1}{2^{n-1}} \xrightarrow{n \rightarrow \infty} 1$$

3. The “Pairs” function truth density is balanced:

$$(Eq. 13) \Rightarrow TD_{Pairs} \sim 1 - \frac{1}{2^1} - \frac{1}{2^{n-1}} \xrightarrow{n \rightarrow \infty} \frac{1}{2}$$

4. The threshold functions truth density is biased towards 1:

$$\begin{aligned} (Eq. 10) \Rightarrow TD_{thres} &= \frac{\sum_{i=1}^{n-1} \left[\binom{n-1}{i} \times \sum_{j=0}^{\min(i,1)} \binom{1}{j} \right]}{2^n} = \frac{\sum_{i=1}^{n-1} \left[\binom{n-1}{i} \times \sum_{j=0}^1 \binom{1}{j} \right]}{2^n} \\ &= \frac{\sum_{i=1}^{n-1} \binom{n-1}{i} \times 2}{2^n} = \frac{2^{n-1} - 1}{2^{n-1}} = 1 - \frac{1}{2^{n-1}} \xrightarrow{n \rightarrow \infty} 1 \end{aligned}$$

Scenario 3 A low activator-to-inhibitor ratio ($1 : n - 1$), where all regulators are inhibitors except one activator, i.e. $m = 1, k = n - 1$.

1. The “AND-NOT” function truth density is biased towards 0:

$$(Eq. 11) \Rightarrow TD_{AND-NOT} \sim \frac{1}{2^{n-1}} \xrightarrow{n \rightarrow \infty} 0$$

2. The “OR-NOT” function truth density is balanced:

$$(Eq. 12) \Rightarrow TD_{OR-NOT} \sim 1 - \frac{1}{2^1} = \frac{1}{2}$$

3. The “Pairs” function truth density is balanced:

$$(Eq. 13) \Rightarrow TD_{Pairs} \sim 1 - \frac{1}{2^{n-1}} - \frac{1}{2^1} \xrightarrow{n \rightarrow \infty} \frac{1}{2}$$

4. The threshold functions truth density is biased towards 0:

$$\begin{aligned} (Eq. 10) \Rightarrow TD_{thres} &= \frac{\sum_{i=1}^1 \left[\binom{1}{i} \times \sum_{j=0}^{\min(i,n-1)} \binom{n-1}{j} \right]}{2^n} = \frac{\sum_{j=0}^{\min(1,n-1)} \binom{n-1}{j}}{2^n} \\ &= \frac{\sum_{j=0}^1 \binom{n-1}{j}}{2^n} = \frac{1 + (n-1)}{2^n} = \frac{n}{2^n} \xrightarrow{n \rightarrow \infty} 0 \text{ L'Hôpital Rule} \end{aligned}$$

References

- [1] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–210, Nov 2002. ISSN 00280836. doi: 10.1038/nature01254.
- [2] Han-Yu Chuang, Matan Hofree, and Trey Ideker. A Decade of Systems Biology. *Annual Review of Cell and Developmental Biology*, 26(1):721–744, Nov 2010. ISSN 1081-0706. doi: 10.1146/annurev-cellbio-100109-104122.
- [3] Rolf Apweiler, Tim Beissbarth, Michael R Berthold, Nils Blüthgen, Yvonne Burmeister, Olaf Dammann, Andreas Deutsch, Friedrich Feuerhake, Andre Franke, Jan Hasenauer, Steve Hoffmann, Thomas Höfer, Peter LM Jansen, Lars Kaderali, Ursula Klingmüller, Ina Koch, Oliver Kohlbacher, Lars Kuepfer, Frank Lammert, Dieter Maier, Nico Pfeifer, Nicole Radde, Markus Rehm, Ingo Roeder, Julio Saez-Rodriguez, Ulrich Sax, Bernd Schmeck, Andreas Schuppert, Bernd Seilheimer, Fabian J Theis, Julio Vera, and Olaf Wolkenhauer. Whither systems medicine? *Experimental & Molecular Medicine*, 50(3):e453, Mar 2018. ISSN 2092-6413. doi: 10.1038/emm.2017.290.
- [4] Bree B. Aldridge, John M. Burke, Douglas A. Lauffenburger, and Peter K. Sorger. Physicochemical modelling of cell signalling pathways. *Nature Cell Biology*, 8(11):1195–1203, Nov 2006. ISSN 1465-7392. doi: 10.1038/ncb1497.
- [5] Melody K. Morris, Julio Saez-Rodriguez, Peter K. Sorger, and Douglas A. Lauffenburger. Logic-based models for the analysis of cell signaling networks. *Biochemistry*, 49(15):3216–3224, Apr 2010. ISSN 00062960. doi: 10.1021/bi902202q.
- [6] Pauline Traynard, Luis Tobalina, Federica Eduati, Laurence Calzone, and Julio Saez-Rodriguez. Logic Modeling in Quantitative Systems Pharmacology. *CPT: Pharmacometrics & Systems Pharmacology*, 6(8):499–511, Aug 2017. ISSN 21638306. doi: 10.1002/psp4.12225.
- [7] Vasundra Touré, Åsmund Flobak, Steven Vercruyse, Anna Niarakis, and Martin Kuiper. The status of causality in biological databases: data resources and data retrieval possibilities to support logical modeling. *Briefings in Bioinformatics*, 2020. doi: 10.1093/bib/bbaa390.
- [8] Rui-Sheng Wang, Assieh Saadatpour, and Réka Albert. Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5):55001, Sep 2012. doi: 10.1088/1478-3975/9/5/055001.
- [9] Luis Mendoza and Ioannis Xenarios. A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theoretical Biology and Medical Modelling*, 3(1):13, Mar 2006. ISSN 17424682. doi: 10.1186/1742-4682-3-13.
- [10] Julio Saez-Rodriguez, Leonidas G Alexopoulos, Jonathan Epperlein, Regina Samaga, Douglas A Lauffenburger, Steffen Klamt, and Peter K Sorger. Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Molecular Systems Biology*, 5(1):331, Jan 2009. ISSN 1744-4292. doi: 10.1038/msb.2009.87.
- [11] Santiago Videla, Julio Saez-Rodriguez, Carito Guziolowski, and Anne Siegel. caspo: a toolbox for automated reasoning on the response of logical signaling networks families. *Bioinformatics*, 33(6):947–950, 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw738.
- [12] Enio Gjerga, Panuwat Trairatphisan, Attila Gabor, Hermann Koch, Celine Chevalier, Franceco Ceccarelli, Aurelien Dugourd, Alexander Mitsos, and Julio Saez-Rodriguez. Converting networks to predictive logic models from perturbation signalling data with CellNOpt. *Bioinformatics*, 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa561.
- [13] Sara Sadat Aghamiri, Vidisha Singh, Aurélien Naldi, Tomáš Helikar, Sylvain Soliman, and Anna Niarakis. Automated inference of Boolean models from molecular interaction maps using CaSQ. *Bioinformatics*, 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa484.

- [14] Åsmund Flobak, Anaïs Baudot, Elisabeth Remy, Liv Thommesen, Denis Thieffry, Martin Kuiper, and Astrid Lægreid. Discovery of Drug Synergies in Gastric Cancer Cells Predicted by Logical Modeling. *PLOS Computational Biology*, 11(8):e1004426, Aug 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004426.
- [15] Barbara Niederdorfer, Vasundra Touré, Miguel Vazquez, Liv Thommesen, Martin Kuiper, Astrid Lægreid, and Åsmund Flobak. Strategies to Enhance Logic Modeling-Based Cell Line-Specific Drug Synergy Prediction. *Frontiers in Physiology*, 11:862, Jul 2020. ISSN 1664-042X. doi: 10.3389/fphys.2020.00862.
- [16] Amel Bekkar, Anne Estreicher, Anne Niknejad, Cristina Casals-Casas, Alan Bridge, Ioannis Xenarios, Julien Dorier, and Isaac Crespo. Expert curation for building network-based dynamical models: a case study on atherosclerotic plaque formation. *Database*, 2018(2018):31, Jan 2018. ISSN 1758-0463. doi: 10.1093/database/bay031.
- [17] Yves Crama and Peter L Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.
- [18] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [19] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, Apr 1982. ISSN 00278424. doi: 10.1073/pnas.79.8.2554.
- [20] Stefan Bornholdt. Boolean network models of cellular regulation: prospects and limitations. *Journal of The Royal Society Interface*, 5, Aug 2008. ISSN 1742-5689. doi: 10.1098/rsif.2008.0132.focus.
- [21] Claudine Chaouiya, Ouerdia Ourrad, and Ricardo Lima. Majority Rules with Random Tie-Breaking in Boolean Gene Regulatory Networks. *PLoS ONE*, 8(7):e9626, Jul 2013. ISSN 19326203. doi: 10.1371/journal.pone.0069626.
- [22] José E. R. Cury, Pedro T. Monteiro, and Claudine Chaouiya. Partial Order on the set of Boolean Regulatory Functions, Jan 2019. arXiv preprint arXiv:1901.07623.
- [23] Archie Blake. Canonical expressions in boolean algebra. *PhD Thesis*, 1937. Department of Mathematics, University of Chicago.
- [24] Stuart A Kauffman. *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA, 1993.
- [25] Marco Gherardi and Pietro Rotondo. Measuring logic complexity can guide pattern discovery in empirical systems. *Complexity*, 21:397–408, Aug 2016.
- [26] Itai Benjamini, Oded Schramm, and David B. Wilson. Balanced Boolean functions that can be evaluated so that every input bit is unlikely to be read. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 244–250, New York, USA, 2005. ACM Press. doi: 10.1145/1060590.1060627.
- [27] Christoph Müssel, Martin Hopfensitz, and Hans A. Kestler. BoolNet — an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, 26(10):1378–1380, May 2010. ISSN 1460-2059. doi: 10.1093/bioinformatics/btq124.
- [28] John Zobolas. Gitsbe format documentation, 2020. Retrieved from <https://druglogics.github.io/druglogics-doc/gitsbe-config.html#gitsbe-format>.
- [29] Aurélien Naldi. BioLQM: A Java Toolkit for the Manipulation and Conversion of Logical Qualitative Models of Biological Networks. *Frontiers in Physiology*, 9:1605, Nov 2018. ISSN 1664-042X. doi: 10.3389/fphys.2018.01605.
- [30] Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, Apr 1960. ISSN 0013-1644. doi: 10.1177/001316446002000104.

- [31] Eirini Tsirvouli, Barbara Niederdorfer, John Zobolas, Touré Vasundra, Åsmund Flobak, and Martin Kuiper. CASCADE - CAncer Signaling CAusality DatabasE, Oct 2020. Retrieved from <https://github.com/druglogics/cascade>.
- [32] J. Richard Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159, Mar 1977. ISSN 0006341X. doi: 10.2307/2529310.
- [33] Mary L. McHugh. Interrater reliability: The kappa statistic. *Biochimia Medica*, 22(3):276–282, Oct 2012. ISSN 13300962. doi: 10.11613/bm.2012.031.
- [34] Albert László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, Oct 1999. ISSN 00368075. doi: 10.1126/science.286.5439.509.
- [35] H. Jeong, B. Tombor, R. Albert, Z. N. Oltval, and A. L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, Oct 2000. ISSN 00280836. doi: 10.1038/35036627.
- [36] Stefan Wuchty. Scale-Free Behavior in Protein Domain Networks. *Molecular Biology and Evolution*, 18(9):1694–1702, Sep 2001. ISSN 1537-1719. doi: 10.1093/oxfordjournals.molbev.a003957.
- [37] Réka Albert. Scale-free networks in cell biology. *Journal of Cell Science*, 118(21):4947–4957, Nov 2005. ISSN 00219533. doi: 10.1242/jcs.02714.
- [38] Maximino Aldana, Enrique Balleza, Stuart Kauffman, and Osbaldo Resendiz. Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology*, 245(3):433–448, Apr 2007. ISSN 00225193. doi: 10.1016/j.jtbi.2006.10.027.
- [39] Raya Khanin and Ernst Wit. How scale-free are biological networks. *Journal of Computational Biology*, 13(3):810–818, Apr 2006. ISSN 10665277. doi: 10.1089/cmb.2006.13.810.
- [40] Anna D. Broido and Aaron Clauset. Scale-free networks are rare. *Nature Communications*, 10(1):1–10, Dec 2019. ISSN 20411723. doi: 10.1038/s41467-019-08746-5.
- [41] Maximino Aldana. Boolean dynamics of networks with scale-free topology. *Physica D: Nonlinear Phenomena*, 185(1):45–66, 2003. doi: 10.1016/S0167-2789(03)00174-X.
- [42] Réka Albert and Albert László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, Jan 2002. ISSN 00346861. doi: 10.1103/RevModPhys.74.47.
- [43] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, Mar 1969. ISSN 10958541. doi: 10.1016/0022-5193(69)90015-0.
- [44] Ilya Shmulevich and Stuart A. Kauffman. Activities and sensitivities in Boolean network models. *Physical Review Letters*, 93(4):048701, Jul 2004. ISSN 00319007. doi: 10.1103/PhysRevLett.93.048701.
- [45] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, Feb 2002. ISSN 1367-4803. doi: 10.1093/bioinformatics/18.2.261.
- [46] Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–4786, Apr 2004. ISSN 00278424. doi: 10.1073/pnas.0305937101.
- [47] Maria I. Davidich and Stefan Bornholdt. Boolean Network Model Predicts Cell Cycle Sequence of Fission Yeast. *PLoS ONE*, 3(2), Feb 2008. ISSN 1932-6203. doi: 10.1371/journal.pone.0001672.
- [48] Andreas Wagner. Evolution of gene networks by gene duplications: A mathematical model and its implications on genome organization. *Proceedings of the National Academy of Sciences of the United States of America*, 91(10):4387–4391, May 1994. ISSN 00278424. doi: 10.1073/pnas.91.10.4387.
- [49] Panos Oikonomou and Philippe Cluzel. Effects of topology on network evolution. *Nature Physics*, 2(8):532–536, Aug 2006. ISSN 17452481. doi: 10.1038/nphys359.

- [50] Lars Kaderali, Eva Dazert, Ulf Zeuge, Michael Frese, and Ralf Bartenschlager. Reconstructing signaling pathways from RNAi data using probabilistic Boolean threshold networks. *Bioinformatics*, 25(17): 2229–2235, Sep 2009. ISSN 1460-2059. doi: 10.1093/bioinformatics/btp375.
- [51] Sam F Greenbury, Iain G Johnston, Matthew A Smith, Jonathan P K Doye, and Ard A Louis. The effect of scale-free topology on the robustness and evolvability of genetic regulatory networks. *Journal of Theoretical Biology*, 267(1):48–61, 2010. ISSN 0022-5193. doi: 10.1016/j.jtbi.2010.08.006.
- [52] John Jack, John F. Wambaugh, and Imran Shah. Simulating Quantitative Cellular Responses Using Asynchronous Threshold Boolean Network Ensembles. *BMC Systems Biology*, 5(1):1–13, Jul 2011. ISSN 17520509. doi: 10.1186/1752-0509-5-109.
- [53] Jorge G.T. Zañudo, Maximino Aldana, and Gustavo Martínez-Mekler. Boolean threshold networks: Virtues and limitations for biological modeling. *Information Processing and Biological Systems*, 11: 113–151, 2011. ISSN 18684394. doi: 10.1007/978-3-642-19621-8_6.
- [54] Roded Sharan and Richard M. Karp. Reconstructing Boolean Models of Signaling. *Journal of Computational Biology*, 20(3):249–257, Mar 2013. ISSN 1066-5277. doi: 10.1089/cmb.2012.0241.
- [55] Aurélien Naldi, Céline Hernandez, Nicolas Levy, Gautier Stoll, Pedro T. Monteiro, Claudine Chaouiya, Tomáš Helikar, Andrei Zinovyev, Laurence Calzone, Sarah Cohen-Boulakia, Denis Thieffry, and Loïc Paulevé. The CoLoMoTo Interactive Notebook: Accessible and Reproducible Computational Analyses for Qualitative Biological Networks. *Frontiers in Physiology*, 9:680, Jun 2018. ISSN 1664-042X. doi: 10.3389/fphys.2018.00680.
- [56] Ronald L Graham, Donald E Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., USA, 2nd edition, 1994. ISBN 0201558025.

End of Thesis

