Biological Computation – Final Project

Avital Yoffe Vasiliev
Shaiel Vistuch

Table of Contents

Part 1

Github Link

Background

Each graph represents the state of the activators and inhibitors (repressors) of the component c in check.

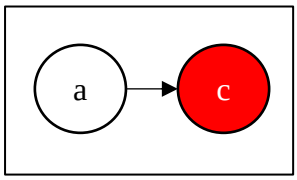


Figure 1: An inactive activator is marked with a white color and an arrow.

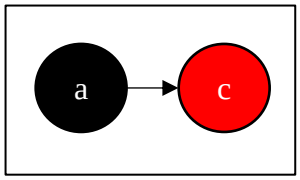


Figure 2: An active activator is marked with a black color and an arrow.

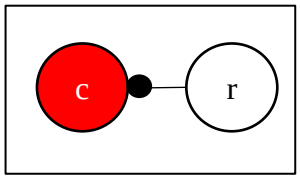


Figure 3: An inactive inhibitor (repressor) is marked with a white color and a line with a circle mark.

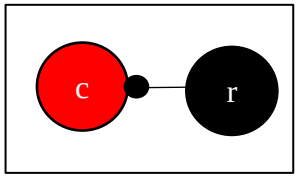
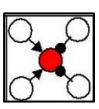
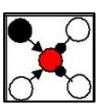
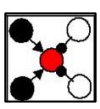
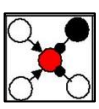
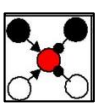
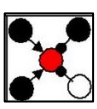
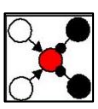
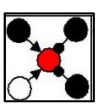
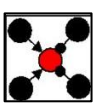


Figure 4: An active inhibitor (repressor) is marked with a black color and a line with a circle mark.

We explained each of the possible combinations in the first row of the following table:

	All activators and repressors aren't present.	Some but not all of the activators are present. None of the repressors are present.	All of the activators are present. None of the repressors are present.	Some but not all of the repressors are present. None of the activators are present.	Some but not all of the represors, as well as some but not all the activators, are present.	Some but not all of the repressors are present. All of the activators are present.	All of the repressors are present. None of the activators are present.	Some but not all of the activators are present. All of the repressors are present.	All activators and repressors are present.
Regulation condition									
0									
1									
2									

3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									

The images of the graphs in this table were adapted from the paper "Modeling the C. elegans Germline Stem Cell Genetic Network using Automated Reasoning" by Ani Amar, E. Jane Albert Hubbard, and Hillel Kugler.

Explanation of the Table:
In the table above, each row corresponds to a specific Boolean function. For each row a red box indicates that the regulated component is active in the state described by the corresponding graph.
For example:

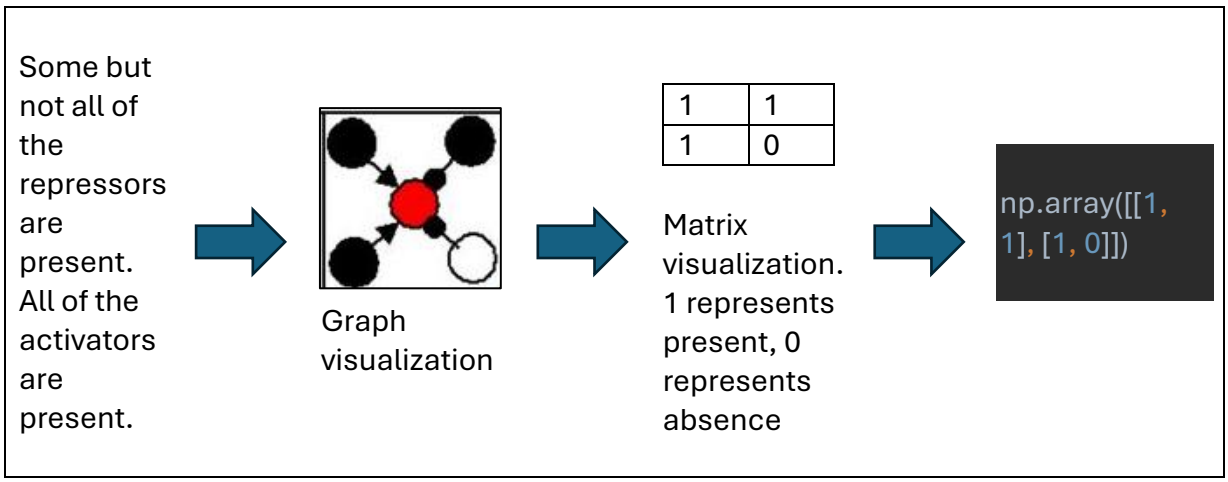
- **Row 0:** The regulated component is active only if *all* activators are present and *none* of the repressors are present.
- **Row 1:** The regulated component is active if either of the following conditions is met:
    1. All activators are present and none of the repressors are present, OR
    2. Some but not all activators are present and none of the repressors are present.

You can analyze each regulation condition in the table in a similar manner to understand the activation rules.

Monotonic property: if a regulated component is active, and one of its activators switches from inactive to active, the regulated component will remain active. Similarly, if a regulated component is inactive, and one of its inhibitors switches from inactive to active, the regulated component will remain inactive.

Code Explanation

First, we created a list corresponding to each state described in the first row of the table. Each list item was made to resemble the original graphs in the table and to accurately represent their meaning.
For example:



The function *generate_possible_functions* returns a list, whose items are all the possible combinations to make a list using just of 0 and 1. The length of each generated 0s-and-1s list is given as a parameter.
The number 1 is used to represent active component, the number 0 is used to represent inactive component. We then must remove the functions [1, 1, 1, 1, 1, 1, 1, 1, 1] and [0, 0, 0, 0, 0, 0, 0, 0, 0], since they correspond to the case were the regulated component remains always active\ always inactive regardless of the state of its activators and repressors.
Afterwards, using the function *find_monotonic_functions* we iterate over each function. For each function, we iterate over its value in each of the states described by the graphs. If we find a situation where the regulated component is active (equal to 1), we also must check if when its activators switches from inactive to active, the regulated component remains active. Similarly, if we find a situation where the regulated component is inactive (equal to 0), we must check if when one of its inhibitors switches from inactive to active, the regulated component remains inactive. The function *find_monotonic_functions* takes the list of all possible functions, as well as the list of all the different states of the activators and repressors, as inputs. It returns a list of all the monotonic functions.
Throughout the iteration over a function, we maintain a Boolean flag variable named *monotonic*. If we find a case where the monotonic property doesn't hold, we change this flag variable from True to False, and break from the sub-loop. If through the entire iteration the flag was not changed, we can infer the function is monotonic and add it to a pre-designated list named *monotonic_functions*. This list is what the function will return eventually.

Running the Code

1. Download the numpy library version 1.26.4 via PyChar Python Packages window or using the command:
   pip install numpy==1.26.4
2. To run the program, simply run the main function.

Results

All the monotonic functions are saved as items of the list *monotonic_functions*.
If we print out the items of the list, and we get:

```
[0, 0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 1, 0, 0, 0]
[0, 0, 1, 0, 0, 1, 0, 0, 1]
[0, 1, 1, 0, 0, 0, 0, 0, 0]
[0, 1, 1, 0, 0, 1, 0, 0, 0]
[0, 1, 1, 0, 0, 1, 0, 0, 1]
[0, 1, 1, 0, 1, 1, 0, 0, 0]
[0, 1, 1, 0, 1, 1, 0, 0, 1]
[0, 1, 1, 0, 1, 1, 0, 1, 1]
[1, 1, 1, 0, 0, 0, 0, 0, 0]
[1, 1, 1, 0, 0, 1, 0, 0, 0]
[1, 1, 1, 0, 0, 1, 0, 0, 1]
[1, 1, 1, 0, 1, 1, 0, 0, 0]
[1, 1, 1, 0, 1, 1, 0, 0, 1]
[1, 1, 1, 0, 1, 1, 0, 1, 1]
[1, 1, 1, 1, 1, 1, 0, 0, 0]
[1, 1, 1, 1, 1, 1, 0, 0, 1]
[1, 1, 1, 1, 1, 1, 0, 1, 1]

Process finished with exit code 0
```

Like mentioned in the previous section, a 1 represents an active component and a 0 represents an inactive component.
We can see that the resulting functions in the list each corresponds to one of the 18 rows in the table:

[0, 0, 1, 0, 0, 0, 0, 0, 0] – row 0
[0, 0, 1, 0, 0, 1, 0, 0, 0] – row 2
[0, 0, 1, 0, 0, 1, 0, 0, 1] – row 4
[0, 1, 1, 0, 0, 0, 0, 0, 0] – row 1
[0, 1, 1, 0, 0, 1, 0, 0, 0] – row 3
[0, 1, 1, 0, 0, 1, 0, 0, 1] – row 5
[0, 1, 1, 0, 1, 1, 0, 0, 0] – row 6
[0, 1, 1, 0, 1, 1, 0, 0, 1] – row 7
[0, 1, 1, 0, 1, 1, 0, 1, 1] – row 8
[1, 1, 1, 0, 0, 0, 0, 0, 0] – row 9
[1, 1, 1, 0, 0, 1, 0, 0, 0] – row 10
[1, 1, 1, 0, 0, 1, 0, 0, 1] – row 13
[1, 1, 1, 0, 1, 1, 0, 0, 0] – row 11
[1, 1, 1, 0, 1, 1, 0, 0, 1] – row 14
[1, 1, 1, 0, 1, 1, 0, 1, 1] – row 16
[1, 1, 1, 1, 1, 1, 0, 0, 0] – row 12
[1, 1, 1, 1, 1, 1, 0, 0, 1] – row 15
[1, 1, 1, 1, 1, 1, 0, 1, 1] – row 17

Overall, there are 18 rows in the original given table and at the code output.

Part 2

Repository of Logically Consistent Real-World Boolean Network Models

Samuel Pastva, David Šafránek, Nikola Beneš, Luboš Brim, Thomas Henzinger

The paper, published on June 12, 2023, introduces the Biodivine Boolean Models (BBM) Benchmark Dataset, a comprehensive dataset of over 210 Boolean network models gathered from various research papers. This dataset is publicly available on GitHub. Boolean networks play a crucial role in modelling biological systems. However, there is a lack of comprehensive, open-source and high-quality datasets. Existing datasets are often limited and contain inconsistencies.

The dataset addresses these challenges by offering extensive model coverage in multiple formats and includes a validation process to ensure each model is logically consistent and correct, it compiles models from various well-known sources (such as CellCollective, Biomodels, and GINsim) and various publication institutes (such as Bioinformatics, Springer Nature, the Royal Society of Chemistry) and making them available in multiple formats.

Each model in the dataset comes with detailed information, including the number of variables, inputs, regulations, and any changes made to the original model, along with the source of the publication.

All of the models in the dataset undergone a high-quality validation pipeline that identifies and corrects logical inconsistencies. Examples of inconsistencies repairs includes modifications to regulations to ensure consistent between the regulatory graph and update functions, or to ensure logical functions match the regulatory graph.

The models in the repository are available in several formats, including JSON, AEON, BN, and SBML. This flexibility allows for custom dataset editions and enhancing the dataset usability for various benchmarking and validation tasks.

In conclusion, this Boolean network model dataset offers a large, high-quality and regulated set of models in different formats, compatible with various Boolean network analysis tools. It supports research and tool development in systems biology through collaborative, open-source GitHub repository.

Chosen Model: COLITIS-ASSOCIATED-COLON-CANCER

The publication

Network modelling reveals the mechanism underlying colitis-associated colon cancer and identifies novel combinatorial anti-cancer targets.
Nature, 2015

Overview of the Article's Main Contributions

The paper provides exploration of the mechanisms underlying colitis-associated colon cancer (CAC) through construction and analysis of a boolean network model.
Firstly, the article presents the development of a knowledge-based network that combines both the extracellular microenvironment and intracellular signalling pathways involved in CAC. This network is built of 70 nodes and 153 edges, representing a wide array of biological interactions that drive the progression from inflammation to cancer. The network model is built on extensive literature and database searches, ensuring that it accurately reflects current scientific understanding.
A significant theoretical contribution of the article is the identification of a core regulatory module within the CAC network. This module, including components such as P53, MDM2, and AKT, is shown to be crucial in promoting the malignant transformation of colon epithelial cells within a pro-tumor inflammatory microenvironment (constant activation of dendritic cells (DCs)). The identification of this module provides new insights into the processes that underlie CAC and offers potential targets for therapeutic intervention.
The practical applications of the paper include the use of dynamic simulations to explore the behaviour of the CAC network under various conditions. These simulations reveal that the constant activation of DCs creates a pro-tumor microenvironment, promoting cell proliferation and inhibiting apoptosis. This finding highlights the importance of the immune microenvironment in cancer development and suggests that targeting DCs could be a good strategy for preventing or treating CAC.
The paper also identifies novel therapeutic targets through in-silico mutation experiments. The research finds that simultaneous targeting of the ceramide pathway and the PI3K/AKT pathway can achieve synergistic anti-cancer effects.
Moreover, the study provides a framework for future research by offering a detailed approach to network construction, simulation, and validation. The authors ensure that their model is accessible and reproducible by providing all necessary details in the supplementary materials, including the logical rules of the network and the results of robustness tests.

Shortcoming in the Article

While the article provides a comprehensive analysis of the mechanisms underlying CAC, there are some areas where it could be improved. One notable shortcoming is the inherent simplification within the boolean network models used in the study. These models operate on binary states, which may not fully capture the complex behaviour of biological systems. This simplification can lead to an oversimplified understanding of the dynamic processes involved in cancer development and progression.

To address this limitation, future research could include more advanced modelling techniques that allow for continuous states and quantitative data. For example, integrating probabilistic models or differential equation-based approaches could provide a more accurate representation of the biological interactions and pathways involved in CAC. This would help in capturing the subtle variations in protein expression levels, gene activity, and other factors that influence cancer progression.
Another area for improvement is the scope of experimental validation. While the study includes some experimental validations to confirm the model's predictions, these validations are limited in scope. Expanding the experimental validation to a wider range of cell lines and in vivo models would enhance the robustness of the findings. Additionally, validating the model's predictions in different stages of CAC development and in diverse genetic backgrounds could provide a more robust understanding of the disease and its potential therapeutic targets.

Impressions of the Article

While our impressions of the article are mostly positive, we believe that additional research could be beneficial.
The research is well-structured and provides significant insights into the mechanisms of CAC. The use of Boolean network modelling to identify potential therapeutic targets is innovative and demonstrates the power of computational approaches in cancer research. The study's ability to integrate a wide array of biological data into a cohesive network model is valuable, as it offers an extensive view of the complex interactions driving CAC.
The identification of novel therapeutic targets, such as the dual-targeting of ceramide and the PI3K/AKT pathway, is a noteworthy achievement that could have significant clinical implications.
However, the article does have some limitations. The simplification inherent in Boolean network models, which only allow for binary states, limits the validity of the insights. Additionally, expanding the experimental work to include a wider variety of cell lines and in vivo models would strengthen the findings and increase their generalizability. It would also be beneficial to validate the model's predictions in different stages of CAC development and across diverse genetic backgrounds.
In summary, while there are areas for improvement, the article is a significant and valuable contribution to the understanding of CAC. The study's innovative approach and attentive methodology make it a notable addition to the field.

Reproducement Of the Main Results

To reproduce the results depicted in figures 2 and 3 of the paper, we utilised the SimpleBool library, along with the rules provided in the supplementary information of the paper. By directly reading the model and parameter input files, SimpleBool facilitated dynamic simulations and attractor identification.

Dynamics of the CAC network model in a normal immune microenvironment

Under non-inflammatory conditions, key nodes such as proliferation, STAT3, NFKB, and BCATENIN stabilised in the OFF state, indicating a reduced likelihood of cell proliferation. In contrast, the apoptosis node displayed an increased activation frequency, suggesting that a significant fraction of epithelial cells were more prone to undergo apoptosis in the absence of inflammatory signals. These findings align with the expected biological behaviour of IECs in a resting state, where the suppression of proliferative activity and the promotion of apoptosis help maintain tissue homeostasis.

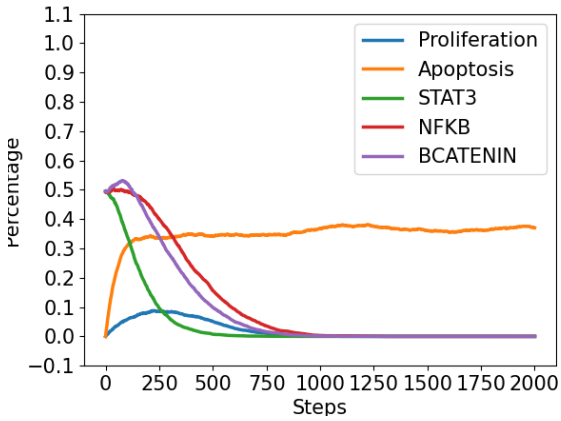


Figure 5: Dynamics in normal immune microenvironment

In the presence of a transiently activated DC, inflammatory response was simulated, showing a increase in the activation frequency of the Proliferation node, while Apoptosis decreases. This led the CAC network to exhibit attractors associated with proliferation (STAT3, NFKB, and BCATENIN were also activated) alongside apoptosis and resting state.

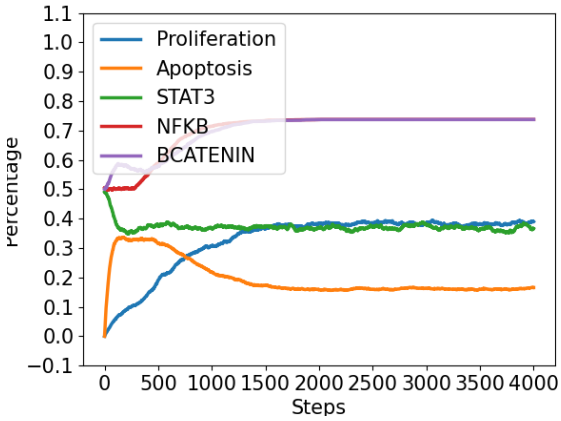


Figure 6: Dynamics of transient DC activation

Dynamics of the CAC network model in a pro-tumour inflammatory microenvironment

Under a pro-tumor inflammatory microenvironment, we observed an increase in proliferation and apoptosis stabilized on OFF. Under these conditions, key nodes such as STAT3, NFKB, and BCATENIN showed heightened activation, reflecting the influence of pro-inflammatory signals that drive tumor-promoting activities. These changes in the network dynamics represent the shift towards a tumor-supportive environment, where the balance between cell growth and death is disrupted, promoting the progression of malignancy.

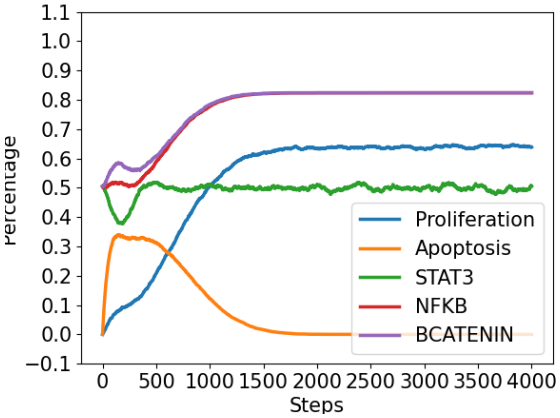


Figure 7: Dynamics in pro-inflammatory microenvironment

Appendices

1. Figure 5 input file

```
rules = rules.txt
ini_on =
ini_off = Proliferation, Apoptosis
turn_on = APC
turn_off = DC,TREG,TGFB,TH1,CTL,IFNG,IL12,TNFA,MAC,IL4,IL10,PGE2,TH2,CCL2,IL6
rounds = 5000
steps = 4000
mode = GA
plot_nodes = Proliferation, Apoptosis, STAT3, NFKB, BCATENIN
missing = random
```

2. Figure 6 input file

```
rules = rules.txt
ini_on = DC
ini_off = Proliferation, Apoptosis
turn_on = APC
turn_off = DC
rounds = 5000
steps = 4000
mode = GA
plot_nodes = Proliferation, Apoptosis, STAT3, NFKB, BCATENIN
missing = random
```

3. Figure 7 input file

```
rules = rules.txt
ini_on =
ini_off = Proliferation, Apoptosis
turn_on = APC, DC
turn_off =
rounds = 5000
steps = 4000
mode = GA
plot_nodes = Proliferation, Apoptosis, STAT3, NFKB, BCATENIN
missing = random
```

4. Rules file:

ATM* = ROS
ASK1* = ROS and not P21
AKT* = PI3K and not (PP2A or CASP3)
BAX* = ((TBID or P53) and PP2A) and not AKT
BCATENIN* = not (GSK3B and APC)

BCL2* = (STAT3 or NFKB) and not (P53 or PP2A)
CASP3* = (CASP8 or CASP9) and not IAP
CASP8* = FADD and not (CFLIP or P21)
CASP9* = CYTC and not (IAP or P21)
CERAMIDE* = SMASE and not SPHK1
CFLIP* = NFKB
COX2* = S1P and TNFR
CYCLIND1* = (BCATENIN or STAT3 or JUN) and not GSK3B
CYTC* = MOMP
EP2* = PGE2
ERK* = MEK
FAS* = CTL
FADD* = TNFR or FAS
FOS* = ERK
GP130* = IL6
GSK3B* = not (EP2 or AKT)
IAP* = (NFKB or STAT3) and not SMAC
IKB* = not IKK
IKK* = (AKT or (S1P and TNFR))
JAK* = GP130 and not SOCS
JNK* = ASK1 or MEKK1
JUN* = ((BCATENIN or ERK) and JNK) and not GSK3B
MDM2* = (P53 and AKT) and not (GSK3B or ATM)
MEK* = RAF or ROS
MEKK1* = CERAMIDE or TGFR or TNFR
MOMP* = (BAX or TBID or CERAMIDE) and not BCL2
NFKB* = not IKB
P21* = (P53 or SMAD) and not (GSK3B or CASP3)
P53* = (PTEN or JNK or ATM) and not MDM2
PGE2* = COX2
PI3K* = (EP2 or RAS) and not PTEN
PP2A* = CERAMIDE and not AKT
PTEN* = P53 and not (NFKB or JUN)
RAF* = CERAMIDE or RAS
RAS* = EP2 or GP130
ROS* = TNFR and not SOD
SOD* = NFKB or STAT3
S1P* = SPHK1
SMAC* = MOMP
SMAD* = TGFR and not JUN
SMAD7* = SMAD or NFKB
SMASE* = P53 or FADD
SPHK1* = ERK or TNFR
STAT3* = JAK
SOCS* = STAT3
TBID* = CASP8 and not BCL2
TGFR* = TGFB and not SMAD7

TNFR* = TNFA
TREG* = (IL10 or DC) and not IL6
TNFA* = MAC
TH2* = IL4 and not (IFNG or TGFB)
TH1* = (IL12 or IFNG) and not (IL10 or TGFB or IL4)
TGFB* = TREG
MAC* = (IFNG or CCL2) and not IL10
IL6* = MAC or DC or NFKB
IL4* = DC or TH2
IL12* = DC or MAC
IL10* = TREG or TH2
IFNG* = TH1 or CTL
CTL* = IFNG and not TGFB
DC* = (CCL2 or TNFA) and not IL10
CCL2* = NFKB
Proliferation* = (FOS and CYCLIND1) and not (P21 or CASP3)
Apoptosis* = CASP3