SSIE 583 Project:
Generating Look-up tables from Partial Input Data in CANA

Srikanth Iyer
Thomas J. Watson College of Engineering and Applied Science, Binghamton University, SUNY, USA
siyer5@binghamton.edu

May 8, 2024

Abstract

This paper discusses the addition of a new feature to the CANA package, which allows the generation of lookup tables from incomplete data. The current implementation of CANA assumes complete input data, but in real-world scenarios, it is often not possible to provide all the necessary inputs. The goal of this project is to extend the capabilities of CANA to handle partial, or incomplete data to facilitate the identification of gaps and contradictions, and evaluation of the statespace of the boolean network from existing knowledge status. The paper also discusses some issues encountered during the implementation and suggests future work, including incorporating bias into the lookup table generation function. Overall, the addition of the ability to generate lookup tables from incomplete data enhances the usability and robustness of the CANA package, making it a valuable tool for studying and analyzing complex systems in computational biology and systems biology.

Introduction

CANA: A Python Package for Quantifying Control and Canalization in Boolean Networks (Correia et al., 2018) is a powerful tool designed to extract, measure, and visualize canalizing redundancy and effective pathways in controlling dynamics present in Boolean network models. It does so with tools such as the 'effective graph' and 'dynamics canalizing map' as well as others to 'uncover minimum sets of control variables', which are important for controlling and manipulating biological systems. This makes CANA a valuable tool for studying and analyzing complex systems in the field of computational biology and systems biology.

While effective graphs and two-symbol schemata allow us to boil down the output rules to its bare essentials and capture relationships hidden in redundancies, this paper will focus on an addition made to the toolkit of CANA- the ability to generate look-up tables from partially described boolean networks. This feature allows users to efficiently interpolate and extrapolate data points, identifying gaps in input data, and potential contradictions in the lookup table.

Input	Output
000	0
001	1
010	0
011	1
100	0
101	1
110	0
111	1

Table 1: Example Boolean Input for Three Inputs

Input	Output
##1	1

Table 2: Alternative view of table: (1) inputs with only Prime Implicants. The '#' symbol is a 'don't care' symbol.

Background

Currently, CANA has two types of inputs that instantiate a Boolean Node in a Boolean Network.

The **lookup table input**- a list of 2-tuples, where the first element is a binary string of length k and the second element is a binary string of length 1. For example, the lookup table input for a node with $k = 2$ would look like this: Table(1)

Prime Implicants (PI): The CANA package also accepts a partial input, where the user can generate a complete lookup table by specifying just the Prime Implicants (PI). See: Table (2). Currently, when instantiating a boolean node in the boolean network, the package will assume an output value of '0' for all unspecified inputs. This is a reasonable assumption, as the lookup table is a complete representation of the node's behavior. Assuming the completeness of the input data allows the package to find Prime Implicants sufficient to generate the entire lookup table. Similarly, Figure 1 is an example of the Prime Implicant inputs for the LFY node in Arabidopsis Thaliana. This generates the lookuptable in Figure 2.

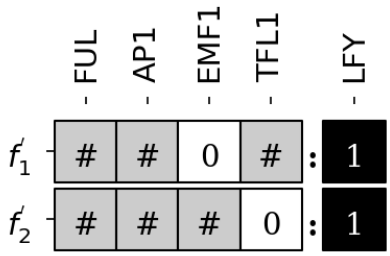

Figure 1: Prime Implicants input (Correia et al., 2018) for the LFY gene in Arabidopsis Thaliana (Chaos et al., 2006). The '#' symbol is a 'don't care' symbol.

Problem Statement

The current implementation of CANA assumes that the lookup table is complete, and the user has provided all the necessary inputs. However, in real-world scenarios, it is not always possible to provide all the inputs. The scientific process is one that is full of uncomfortable interactions with reality- where models and abstractions have to perform in the arena of truth. The truth is not only stern, but also oft elusive. Our understanding of interactions between nodes within bio-regulatory networks relies heavily on rigorous scientific experimentation. These hard-won insights provide the foundation for Boolean Networks, which can then be used to process information and advance our knowledge of these complex systems. It also means that the data we have is often incomplete, noisy, or uncertain. Generating lookup tables from partial data will give us a better understanding of the data we have, what it tells us, and what it doesn't. It will also help us identify gaps in our data, and potentially identify contradictions in the data we have.

Generating Tables from Incomplete Data

The goal of this project is to extend the capabilities of CANA to generate lookup tables from the partial data available to the user. To do this, we will need to:

- Make no assumptions about the completeness of the data. This is done by assigning a value of '?' instead of '0' to all unspecified inputs.
- Parsing input for 'don't care' symbols ('#','–', etc). Iterating all combinations of the boolean values for each symbol and assigning the provided output value to them.
- Identify gaps in the data provided by the user.
- Generating missing input values, assigning the '?' output to them. This will communicate the information-completeness of the boolean network to the user.
- Identify contradictions in the data provided by the user. The current implementation of CANA assumes that there are no contradiction in the data- due to the completeness assumption of the inputs. Contradictory outputs are flagged with a '!' symbol.
- Generate the lookup table from the data provided by the user.

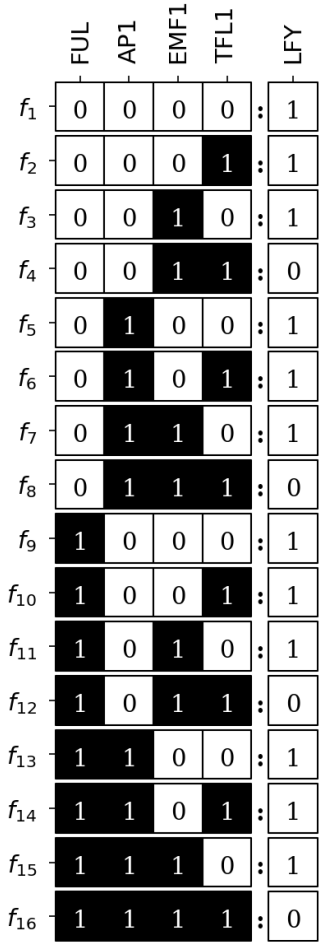

Figure 2: The complete lookup table for the LFY gene in Arabidopsis Thaliana.

With incomplete inputs, including 'don't care' symbols such as the example in Table 3, the lookup table generated by CANA will look like the one in Figure 3.

Issues Encountered and Future Work

While this addition to the CANA package is a step in the right direction, there are still some issues that need to be addressed. With every addition to the package, the probability of unexpected breakages increases.

- The current implementation doesn't account for Prime Implicants as inputs. This is a significant oversight on my part, as the package already has the capability to generate

Input	Output
00##	0
1##1	1
11##	0

Table 3: Incomplete and contradictory input example. The '#' symbol represents 'don't care' inputs.

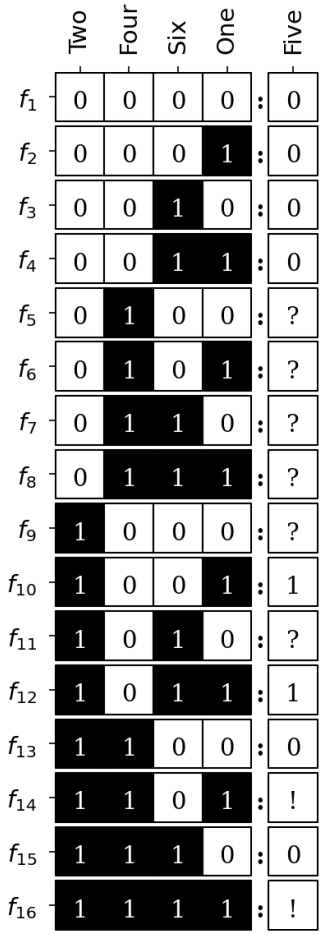

Figure 3: The complete lookup table for Example input from Table 3. The '?' indicates missing input data. The '!' indicates contradictory output values for the given (or permuted) input.

lookup tables from Prime Implicants and the new addition depreciates this feature. I will need to address this issue in the next iteration of the package.

- The styles of input in the datasets in the package vary, which makes discerning the intentions behind the data styles a challenge for me. Evaluating the input txt files to make it more parsable: either in json or csv format can be a potential solution to this issue.
- To distinguish between inputs that are complete and in the form of Prime Implicants and inputs that are incomplete will require further clarification either from the user or in the dataset documentation itself. As of now, deducing this purely from the input values given is proving a challenge. This could be either due to the lack of understanding, or coding inexperience, or it could be an issue that needs to be addressed in future implementations of CANA for the sake of internal consistency of the package and intuitive usability. Ideally, there should be a way to deduce and distinguish both types of data without requiring the user to specify it. This will be my immediate focus on the project.
- Investigating all other potential breakages in the package will be the next step. This includes looking and for more edge cases and testing the package with more datasets .
- Applying Bias to the lookup table generation function will enable the user to generate lookup tables that are more in line with their expectations.

Conclusion

The addition of the ability to generate lookup tables from incomplete data is a useful addition to the CANA package. It allows users to better understand the data they have, identify gaps in their data, and potentially identify contradictions in the data they have. It will also allow the ability to evaluate the statespace of the boolean network, and identify the states that are unreachable, or likely from the given partial data. Incorporating more features such as biasing the generated table, and automatically distinguishing between Prime Implicants and incomplete data will enable the package to be more user-friendly, intuitive, and robust. This will be the focus of the next iteration of the package.

Acknowledgements

I would like to thank Dr. Rocha for the project idea. Learning CANA facilitates learning more about boolean networks and graphs, which is a good foundation for understanding complex systems. I would also like to thank Dr Rozum for handholding me through the CANA package, and his suggestions for future implementation. Thank you for your time and consideration. This work is nowhere near completion and I'm looking forward to finishing it in the near future.

References

Chaos, A., Aldana, M., Espinosa-Soto, C., de Leon, B. G. P., Arroyo, A. G., and Alvarez-Buylla, E. R. (2006). From genes to flower patterns and evolution: Dynamic models of gene regulatory networks. *Journal of Plant Growth Regulation*, 25(4):278–289.

Correia, R. B., Gates, A. J., Wang, X., and Rocha, L. M. (2018). Cana: A python package for quantifying control and canalization in boolean networks. *Frontiers in Physiology*, 9:1046.