Towards a Partitioning of the Input Space of Boolean Networks: Variable Selection Using Bagging

Frank Emmert-Streib[1] and Matthias Dehmer[2]

[1] Queen's University Belfast, Computational Biology and Machine Learning,
Center for Cancer Research and Cell Biology, School of Medicine,
Dentistry and Biomedical Sciences, 97 Lisburn Road, Belfast BT9 7BL, UK
v@bio-complexity.com

[2] Institute for Bioinformatics and Translational Research, UMIT,
Eduard Wallnoefer Zentrum 1, 6060, Hall in Tyrol, Austria
Matthias.Dehmer@umit.at

Abstract. In this paper we present an algorithm that allows to select the input variables of Boolean networks from incomplete data. More precisely, sets of input variables, instead of single variables, are evaluated using mutual information to find the combination that maximizes the mutual information of input and output variables. To account for the incompleteness of the data bootstrap aggregation is used to find a stable solution that is numerically demonstrated to be superior in many cases to the solution found by using the complete data set all at once.

Keywords: Bootstrap aggregation, Mutual Information, Boolean networks, Causality.

1 Introduction

The analysis of networks and their inference has gained much attention during the last years. This interest is at least twofold. First, networks are very interesting objects from a mathematical point of view that possess a multitude of properties that are still to be investigated [1,5,6,16,22,21]. Second, networks can serve as representation of phenomena, e.g., from physics, chemistry or biology [11] to allow their systematic investigation. Especially, in molecular biology networks are nowadays found omnipresently representing, e.g., signaling, metabolic or protein networks [2,3,12,17]. It is important to emphasize that in many of the cases mentioned above networks represent some form of 'interaction' occurring within the system. That means the network structure represents causal dependencies or independencies among the variables in the system [9,19]. For this reason, the inference of, e.g., gene networks from experimental data represents one of the major goals in molecular biology because the inferred networks allow to gain insights in the causal working mechanism of living cells.

In this paper, we present an information-theoretic method that allows to identify a set of input variables of a Boolean network. That means, we are aiming to

J. Zhou (Ed.): Complex 2009, Part I, LNICST 4, pp. 715–723, 2009.

identify a set of variables that effects, potentially causally, the outcome of another variable. We use Boolean networks because gene networks are frequently modeled as Boolean networks [13] and the inference of gene networks is an application we have in mind when designing our method. The algorithm we suggest is based on a recent method by Liang et al. [14] which they called REVEAL (Reverse Engineering Algorithm). We extend this algorithm regarding two important points. First, we modify the algorithm that it can also deal with incomplete data. Second, we use bagging (bootstrap aggregation) to find the optimal input set that is more robust against noise or outliers in the data [7]. We want to emphasize that such a selection mechanism is useful with respect to the partiti oning of the input space of Boolean networks because it may allows to reduce the complexity of post-processing steps. Further, the obtained sets of variables can be seen as larger components (larger than single variables) that might be used to construct the overall network implying approaches that are beyond node-to-node based tests like d-separation [10,18,20] or ARACNE [4,15].

This paper is organized as follows. In the next two sections, we present our method and in section 3 we present numerical results. This paper finishes in section 4 with conclusions.

2 Methods

For our study we assume that we have a given Boolean network that is defined via its lookup table (LUT). The LUT provide a mapping from the binary input variables to the binary output variables. For a Boolean gate with n input variables there is a total of 2^n different combinations that can be realized by n binary variables. Hence, a complete LUT consists of 2^n entries. The method we propose is intended to be used for an incomplete LUT. That means, only a certain fraction of all possible input combinations is observed and used as training set to identify the set of input variables.

Our method consists of a modified version of the REVEAL algorithm. The principle idea of the REVEAL algorithm [14] is that under ideal conditions the entropy of an output is completely determined by the mutual information between the output and its input I_s, i.e.,

$$H(O) = I(O; I_s). \tag{1}$$

Because the mutual information can be written as

$$I(O; I_s) = H(O) - H(O|I_s) = H(O) + H(I_s) - H(O, I_s) \tag{2}$$

this implies that under ideal conditions

$$H(I_s) = H(O, I_s) \tag{3}$$

holds. The REVEAL algorithm is an iterative algorithm starting with an input I_s consisting of just one variable. If condition 1 does not hold combinations of

Algorithm 1. Maximization of mutual information

```
$d^* = 0$
$I_s^* = \{\}$
**for all** allowed input sets **do**
    update $I_s$
    calculate $H(O)$
    calculate $I(O;I_s)$
    $d = I(O;I_s)/H(O)$
    **if** $d > d^*$ **then**
        $d = d^*$
        $I_s^* = I_s$
    **end if**
**end for**
```

variables are used as input I_s until the perfect configuration is found. If no perfect solution is found the algorithm does not make any suggestion for a candidate set.

In this paper we extend the algorithm above by allowing data that are not sufficient for a perfect recovery of input variables, e.g., due to noise. This corresponds to more realistic situations because, e.g., experimental data will always contain noise to some extend that counteracts our goal to identify the perfect input set of an output variable. For this reason we need to modify the optimality criteria in Eq. 1 to account for experimental data in general. This can be accomplished by two modifications. First, note that entropies are always non-negative [8]. This implies that (from Eq. 2)

$$I(O;I_s) \leq H(O), \tag{4}$$

holds. Hence, we are searching an input set that maximizes the fraction, i.e.,

$$I_s^* = \underset{I_s}{\text{argmax}} \left\{ \frac{I(O;I_s)}{H(O)} \right\}. \tag{5}$$

Here I_s^* corresponds to the optimal input set that can be found for given data from which all entropies are estimated. The second modification consists in the stopping criteria. Because we do no longer expect to find a perfect solution we calculate $I(O;I_s)$ for all input sets we want to consider. It is clear that the number of possible input sets I_s increases rapidly with the number of available inputs for this reason we restrict this complexity by selecting only a subset thereof. More precisely, in this study we allowed only input sets of size up to $3 = |I_s|$. Here $|.|$ measures the cardinality of the set I_s. Algorithm 1 gives pseudo code of the principle mechanism of our approach.

In addition to these two modification we use bagging (bootstrap aggregation) [7] to produce a more stable result. Briefly, we sample B bootstrap samples from the original data of the same size (with replacement) and obtain this way B solution sets I_s^b, $b \in \{1, \ldots, B\}$. Hereby we consider each possible input set I_s as a model m. For these models we calculate the probability p_m that model m has

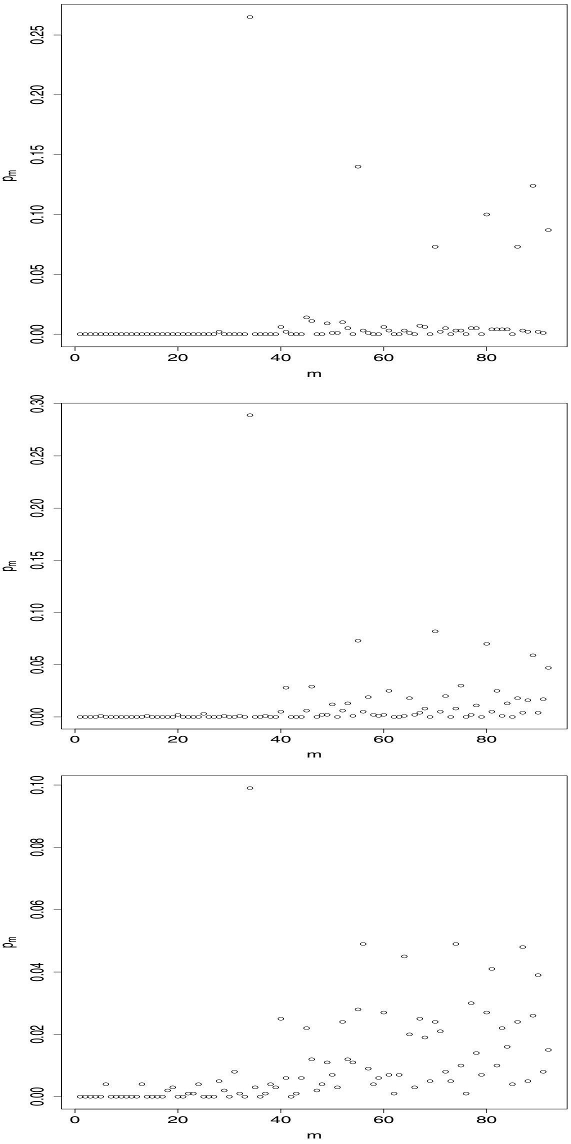

Fig. 1. Results for node 8. Top: $N_s = 77$ (30%) with $V_B = \{6, 7\}$, $V_T = \{1, 5\}$. Middle: $N_s = 51$ (20%) with $V_B = \{6, 7\}$, $V_T = \{5\}$. Bottom: $N_s = 25$ (10%) with $V_B = \{6, 7\}$, $V_T = \{3\}$.

been chosen by all B bootstrap samples. This give us finally the model, input set, that gives the most stable result,

$$V_B = m^* = \underset{m}{\operatorname{argmax}}\{p_m\}. \tag{6}$$

We apply bagging to obtain a probability distribution over all possible models. This allows us in addition to obtain an optimal solution for given data to see how probable other models are for the same data.

3 Results

In this paper we study a Boolean network consisting of 8 variables. The network is defined by the following equations corresponding to a synchronous updating (as in [14])

$$O_1 = I_1 \tag{7}$$
$$O_2 = I_2 \tag{8}$$
$$O_3 = I_3 \tag{9}$$
$$O_4 = I_4 \tag{10}$$
$$O_5 = I_5 \tag{11}$$

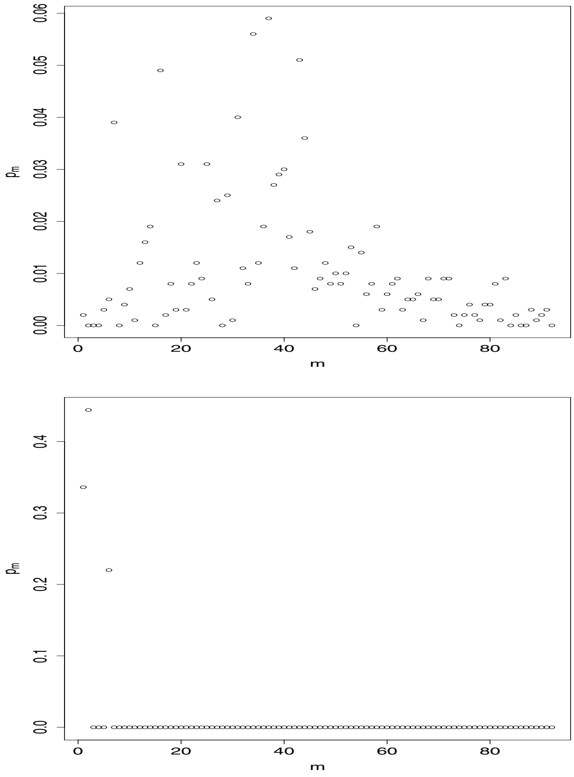

Fig. 2. Results for node 8. Top: $N_s = 13$ (5%) with $V_B = \{1, 2, 3\}$, $V_T = \{1, 3, 8\}$. Bottom: $N_s = 3$ (1%) with $V_B = \{2\}$, $V_T = \{1\}$.

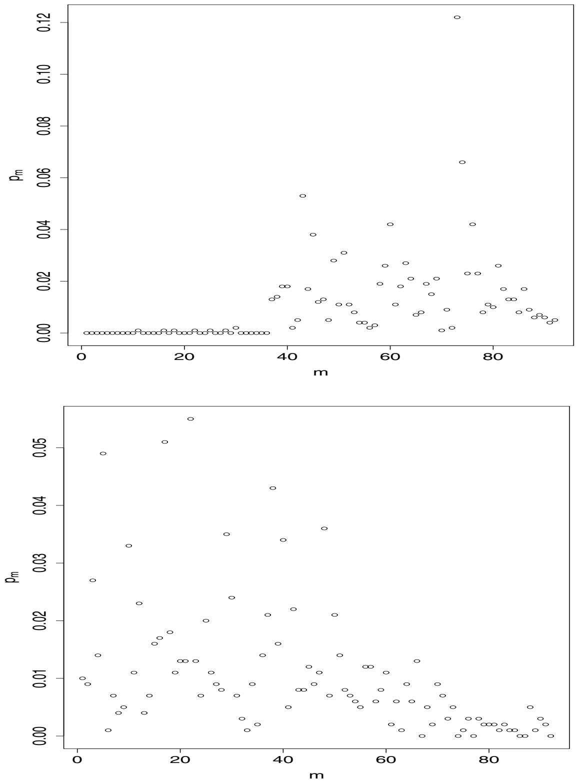

Fig. 3. Results for node 7. Top: $N_s = 25$ (10%) with $V_B = \{3, 4, 5\}$, $V_T = \{3, 7, 8\}$. Bottom: $N_s = 13$ (5%) with $V_B = \{3, 4\}$, $V_T = \{4, 6\}$.

$$O_6 = I_1 \text{ or } I_2 \tag{12}$$

$$O_7 = D(I_3, I_4, I_5) = (I_3 \text{ and } I_4) \text{ or } (I_3 \text{ and } I_5) \text{ or } (I_4 \text{ and } I_5) \tag{13}$$

$$O_8 = I_6 \text{ or } I_7 \tag{14}$$

Here 'and' and 'or' correspond to logical gates and 'D' is defined by the right hand side of Eqn. 13. Each input (I) or output (O) variable can assume values in $\{0, 1\}$.

The purpose of our algorithm introduced in section 2 consists in finding input sets for the output variables of a Boolean network that form at least candidates for a causal dependence among these variables.

For our analysis we use always $B = 1000$ bootstrap samples from given data. This implies that there are $256 = 2^8$ state transitions. Because we are aiming to realistic situations we use just a fraction of all possible states. In Figure 1 we show results for O_8. p_m gives the probability that model m has been selected by the bootstrap samples. For example, for the top figure $N_s = 77$ state transitions (randomly sampled from all 256) have been used as data. All of these results

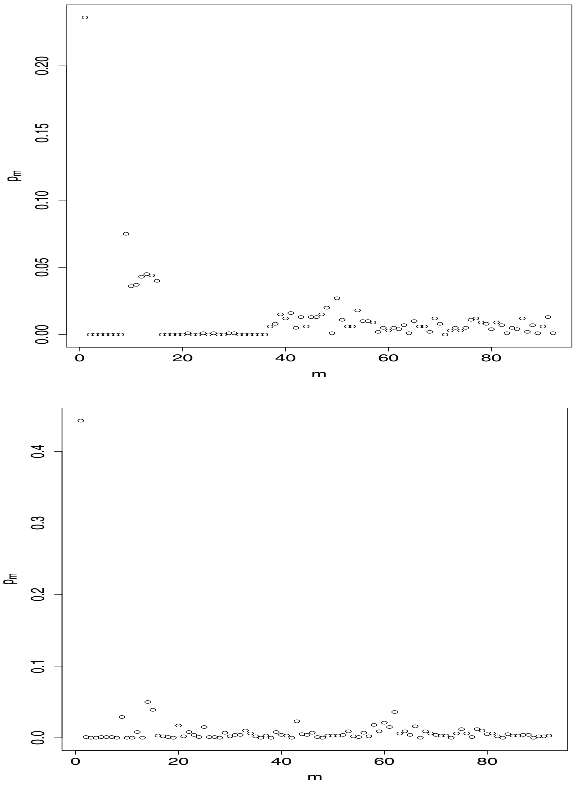

Fig. 4. Results for node 1. Top: $N_s = 25$ (10%) with $V_B = \{1\}$, $V_T = \{1\}$. Bottom: $N_s = 13$ (5%) with $V_B = \{1\}$, $V_T = \{1\}$.

V_B find the correct input set $\{I_6, I_7\}$. Even if we use only $N_s = 25$ samples (bottom figure), corresponding to 10% of all state transitions, gives the correct result. Instead, when we use all data, not applying bagging, we find in all three cases suboptimal input sets V_T. Further reducing the amount of data finally results also in suboptimal input sets as can be seen in Fig. 2. An interesting observation from Fig. 1 and 2 is that the probability distribution over all possible models involving three or less variables is shifted from the right to the left [1]. The interpretation for this shift is that the less data are available (low N_s values) the less complex (lower number of variables) is the input set our algorithm suggests. This behavior is reasonable because less data allow only simpler models without risking over-fitting the data. Figure 3 and 4 also show this behavior. We repeated our analysis 100 times drawing new test data of size $N_s = 25$ (10% of the data) to study the behavior of the population. We found that in 70% of the cases our algorithm identifies the correct input set.

[1] The models are enumerated from simple (left) to more complex sets (right) comprising more variables.

In figure 3 and 4 we show two further results for O_7 and O_1. Also for these two cases 10% of the data seem to be sufficient to identify the correct input set. Again, using all data without bagging gives worse results, only the simple case $O_1 = I_1$ can be identified correctly. This demonstrates the usefulness of bagging and justifies its use in our algorithm. Simulation results of further Boolean networks containing different logical gates for varying parameters of our algorithm and the number of data points confirm our results demonstrating that the exemplary results presented in this section that visualize the working mechanism of our algorithm hold in general. Also for node 7 we studied the population behavior by drawing 100 times new data of size $N_s = 25$ (10% of the data). Here we found that in 62% of the cases the correct input set could be found.

4 Conclusions

In this paper we presented an extension of the REVEAL algorithm [14]. The extended algorithm provides a probability for each allowed input set (model) of a Boolean network by using bagging. We demonstrated that for incomplete lookup tables bagging gives better results than by using all data at once. In general, the probabilistic evaluation of all allowed models could be exploited by ranking all models to identify the candidate models that explain the entropy of the output variable sufficiently well. Here 'sufficiently well' could be quantified by significance tests based on, e.g., the randomization of the data. Our approach represents a first step towards such a realization containing important ingredients that would allow to partition the input space of Boolean networks according to the significance of input sets rather than single variables. In this article we just focused on the most probable set. In future work we will study the more general problem.

Acknowledgements

M. D. thanks the Ciência 2006 fund (Portugal) for financial support.

References

1. Albert, R., Barabasi, A.: Statistical mechanics of complex networks. Rev. of Modern Physics 74, 47 (2002)
2. Alon, U.: An Introduction to Systems Biology: Design Principles of Biological Circuits. Chapman & Hall/CRC, Boca Raton (2006)
3. Barabasi, A.L., Oltvai, Z.N.: Network biology: Understanding the cell's functional organization. Nature Reviews 5, 101–113 (2004)
4. Basso, K., Margolin, A.A., Stolovitzky, G., Klein, U., Dalla-Favera, R., Califano, A.: Reverse engineering of regulatory networks in human b cells. Nature Genetics 37(4), 382–390 (2005)
5. Bollobas, B.: Modern Graph Theory. Springer, Heidelberg (1998)

6. Bornholdt, S., Schuster, H. (eds.): Handbook of Graphs and Networks: From the Genome to the Internet. Wiley, Chichester (2003)
7. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
8. Cover, T., Thomas, J.: Information Theory. John Wiley & Sons, Inc., Chichester (1991)
9. Cox, D., Wermuth, N.: Multivariate dependencies: Models, analysis and interpretation. Chapman & Hall/CRC, Boca Raton (1996)
10. de la Fuente, A., Bing, N., Hoeschele, I., Mendes, P.: Discovery of meaningful associations in genomic data using partial correlation coefficients. Bioinformatics 20(18), 3565–3574 (2004)
11. Dehmer, M., Emmert-Streib, F. (eds.): Analysis of Complex Networks: From Biology to Linguistics. Wiley-VCH, Chichester (in press, 2009)
12. Jeong, H., Mason, S.P., Barabasi, A.L., Oltvai, Z.N.: Lethality and centrality in protein networks. Nature 411, 41–42 (2001)
13. Kauffman, S.: Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology 22, 37–467 (1969)
14. Liang, S., Fuhrman, S., Somogyi, R.: Reveal, a general reverse engineering algorithm for inference of genetic network architectures. In: Pac. Symp. Biocomput., pp. 18–29 (1998)
15. Margolin, A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., Califano, A.: Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7, S7 (2006)
16. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)
17. Palsson, B.: Systems Biology. Cambridge University Press, Cambridge (2006)
18. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)
19. Pearl, J.: Causality: Models, Reasoning, and Inference, Cambridge (2000)
20. Shipley, B.: Cause and Correlation in Biology. Cambridge University Press, Cambridge (2000)
21. Watts, D.: Small Worlds: The Dynamics of Networks between Order and Randomness. Princeton University Press, Princeton (1999)
22. Watts, D., Strogatz, S.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)