Inferring Boolean Networks from Single-Cell Human Embryo Datasets

Mathieu Bolteau, Jérémie Bourdon, Laurent David, Carito Guziolowski

Inferring Boolean Networks from Single-Cell Human Embryo Datasets

Mathieu Bolteau[1], Jérémie Bourdon[1], Laurent David[2], and Carito Guziolowski[1]

[1] Nantes Université, Ecole Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000, Nantes, France
{mathieu.bolteau, jeremie.bourdon, carito.guziolowski}@ls2n.fr
[2] Nantes Université, CHU Nantes, Inserm, CR2TI, F-44000, Nantes, France
laurent.david@univ-nantes.fr

Abstract. This study aims to understand human embryonic development and cell fate determination, specifically in relation to trophectoderm (TE) maturation. We utilize single-cell transcriptomics (scRNAseq) data to develop a framework for inferring computational models that distinguish between two developmental stages. Our method selects pseudo-perturbations from scRNAseq data since actual perturbations are impractical due to ethical and legal constraints. These pseudo-perturbations consist of input-output discretized expressions, for a limited set of genes and cells. By combining these pseudo-perturbations with prior-regulatory networks, we can infer Boolean networks that accurately align with scRNAseq data for each developmental stage. Our publicly available method was tested with several benchmarks, proving the feasibility of our approach. Applied to the real dataset, we infer Boolean network families, corresponding to the medium and late TE developmental stages. Their structures reveal contrasting regulatory pathways, offering valuable biological insights and hypotheses within this domain.

Keywords: Boolean networks · Answer Set Programming · Human preimplantation development · scRNAseq modeling.

1 Introduction

One of the outstanding questions of the field of in vitro fertilization is to understand the chain of events regulating human preimplantation development leading to an implantation-competent embryo. To address this question, in [9], we analyzed single-cell transcriptomic data (scRNAseq) from preimplantation human embryos. Our analysis proposed some hierarchy of transcription factors in epiblast, trophectoderm and primitive endoderm lineages. Individual cell fate within heterogeneous samples, such as human embryos, can be followed from scRNAseq data but presents multiple computational challenges with normalization and "zero-inflation", complicating network models [7]. The state-of-the-art

tools used to propose a temporal distribution of such data are based on statistical approaches, such as manifolds (UMAP [8]) or graphs theory (pseudo-time [11]). In [3], the authors used such pseudo-time distributions along with scRNAseq expression data to infer Boolean networks for modeling gene regulation in cancer progression. They focus, however, on the hypothesis of an averaged cell expression at each stage defined by pseudo-time analysis, allowing to model the dynamics of the cell fate decision. In the context of embryo development, Dunn *et al.* [4] proposed computational models on transcriptional networks from knockout data on mouse stem cells. This data type is ideal since the proposed perturbations add crucial information to the inferring process.

In this work, we propose a framework to discover a family of Boolean networks (BNs) of human preimplantation development that capture the progression from one developmental stage to the next. This framework uses prior-knowledge networks (PKN) as a base on which the scRNAseq data is mapped. Then, it identifies *pseudo-perturbations* specific for two developmental stages. These pseudo-perturbations are used in the last step to infer stage-specific BNs models. Since perturbation data is rarely available due to practical and legal concerns, our main contribution was to extract pseudo-perturbation data from scRNAseq data, considering its high redundancy and sparsity. We used the Pathway Commons database [12] to build a PKN and discovered 20 pseudo-perturbations (across 10 genes) characterizing medium and late stages of trophectoderm (TE) maturation. They correspond to the gene expression of 20 cells in each stage; representative on average of 82% of the total cells. Pseudo-perturbations referring to 10 (entry) genes expression were connected (PKN information) to 14 genes (output) expression. The 20 entry-output gene expression configurations allowed us to infer 2 families of BNs (composed of 8 and 15 logical gates) characterizing medium and late TE developmental stages.

2 Method

2.1 Pipeline overview

Our pipeline is based on background notions stated in Appendix and its main steps, illustrated in supplementary material[3], are: (*i*) PKN reconstruction, (*ii*) experimental design construction, and (*iii*) BNs inference.

PKN reconstruction is achieved by querying the Pathway Commons database, using pyBRAvo [6], with an initial gene list. Briefly, given a list of genes relevant to the case study, pyBRAvo explores recursively predecessors genes and outputs a signed-directed graph. The reconstructed PKN is then reduced to only include genes and their interactions measured in the scRNAseq data, as well as protein complexes associated with the genes to maintain their connectivity. The resulting PKN comprises nodes selected as input, intermediate, and readout.

[3] https://github.com/mathieubolteau/scRNA2BoNI/tree/master/ISBRA_2023_Supp

Experimental design construction This step constructs an experimental design from the reduced PKN and the scRNAseq data of the two studied cell classes (see an example in supplementary material). The experimental design is composed of: (*i*) pseudo-perturbations, which are binarized expression values for input and intermediate genes in chosen cells whose value is identical in both cell classes, and (*ii*) readout observations, which are normalized expression values for readout genes in the chosen cells of both cell classes. To capture the diversity of genes expression in scRNAseq data for each class, we implement a logic program to maximize the number of different pseudo-perturbations for k genes, given a set of input and intermediate genes (see Section 2.3). The resulting experimental design is based on the inputs, intermediates, and readouts of the PKN obtained in the previous step.

BN inference We infer BNs for each class using Caspo [13]. Given a PKN and an experimental design, Caspo learns a family of BNs compatible with the network's topology and the experimental design data. Caspo learns minimal (in size) BNs which minimize the error between their readouts predictions and experimental measures. In our framework, Caspo proposes specific BNs for each class. This is obtained thanks to the experimental design identified in the previous step, where a maximal number of entry-output associations is proposed with common entry gene values in both classes (pseudo-perturbations), and (maximally) different output gene values.

2.2 Experimental data preprocessing

We used single-cell data from [10], which measures the expression of $\sim 20,000$ genes across 1529 cells. Since we focused on genes in the PKN, our dataset comprised 125 genes (111 input and intermediates, and 14 readouts). We considered only cells at medium and late TE stage; therefore we had a total of 680 cells.

First, we discretize raw gene expression data of input and intermediate PKN nodes (see Section 2.1, PKN reconstruction) by considering a gene expressed if at least 2 reads are identified in the raw data. Here, we denote by e_{ij} (resp. r_{ij}) is the binarized (resp. raw) expression of the gene j for the cell i. We have $e_{ij} = 0$ if $r_{ij} < 2$, and $e_{ij} = 1$ otherwise.

Second, we normalize the raw expression of genes related to PKN readouts (see Section 2.1). We denote by n_{ij} the normalized expression of the gene i for the cell j. We have $n_{ij} = (r_{ij} - min)/(max - min)$ where min (resp. max) is the minimum (resp. maximum) expression value of all readout genes across all cells.

2.3 Experimental design construction - algorithm

This algorithm receives an integer k, as a parameter, limiting the number of genes to be selected. Its input data is the preprocessed scRNAseq matrix for

input, intermediate, and readout PKN genes. The algorithm retrieves (i) a maximal number of pseudo-perturbations, which identify cells associations between two classes holding identical expression values for a set of k genes, and (ii) cell associations which maximize the readout difference across (redundant) cells associations. The details of this algorithm are presented below.

Maximizing the number of pseudo-perturbations The input of this method is a binary matrix, E, where e_{ij} represents the presence (or activity level) of gene j for cell i (see Section 2.2). The output is a subset of genes and cells that adhere to various constraints, ensuring their pseudo-perturbations are balanced between the different classes (see supplementary material, experimental design example). Let us denote by C, the complete set of cells; and by G, the complete set of genes in our experimental data. Each cell is uniquely associated with one class (either A or B); $C = A \uplus B$. We use the binary matrix, E, to define the relation I^G, $I^G(c_i) = \{g_j \in G | e_{ij} = 1\}$. $I^G(c_i)$ thus represents the active genes, belonging to G, for cell c_i. If $G' \subset G$, then the restriction of I^G to G' is defined by $I^{G'}(c_i) = I^G(c_i) \cap G'$.

Problem formulation. Given an association matrix E, associating a set G of genes to a set C of cells, where C is composed of cells belonging to 2 disjoint sets (classes) A and B; and given a parameter k limiting the number of selected genes, find a subset G' of genes and the largest subset C' ($C' = A' \uplus B' \subset C$, where $A' \subset A$ and $B' \subset B$) satisfying the three following constraints:

1. The size of G' is fixed to k (parameter). For large instances $k << |G|$.
2. $\forall c_1, c_2 \in A'$ (resp. B'), $c_1 \neq c_2$, we verify that $I^{G'}(c_1) \neq I^{G'}(c_2)$.
3. $\forall c_1 \in A'$ (resp. B'), $\exists c_2 \in B'$ (resp. A'), such that we verify $I^{G'}(c_1) = I^{G'}(c_2)$.

From this result, for each $c_i \in C'$ we define a binary vector b^i, such that for $j \in \{1, \cdots, k\}$, $b^i_j = 1$ (resp. $b^i_j = 0$) if gene $g_j \in I^{G'}(c_i)$ (resp. $\notin I^{G'}(c_i)$). b^i is called a pseudo-perturbation. Notice that since the sets G' and C' are not unique, there may exist several pseudo-perturbations vectors.

Constraints justification. The imposed constraints are crucial in light of the entire framework, which handles Boolean network inference and single-cell data. *Constraint 1* reduces the search space, improves computational efficiency, and simplifies the subsequent step of learning Boolean networks. *Constraint 2* prevents redundancy in gene selection from different cells within the same class. This is essential due to the abundance of zero values and redundancy in single-cell data. *Constraint 3* promotes similarity in gene expression values between the two distinct classes. This alignment enables meaningful comparative analysis during the subsequent step of Boolean network inference. Despite the inherent evolutionary differences between cells belonging to different classes, selecting genes with similar expression values allows us to impose comparable entry conditions on the system, facilitating accurate modeling of the distinct regulatory

mechanisms at play. Finally, selecting a larger number of pseudo-perturbations provides more information, enriching the Boolean network inference step and allowing for exploring various regulatory mechanisms.

Maximizing readout difference Pseudo-perturbations identified by the previous algorithm relate cells in A' to those in B'. However, different cell relations may exist for the same pseudo-perturbation vector.

Problem formulation. Given a set of pseudo-perturbation binary vectors, O, and given the matrix of preprocessed scRNAseq data of normalized readout values, find the sets of cells A'^* and B'^*, associated by all pseudo-perturbation vectors in O, that maximize the difference of readout vectors, $r^{A'*}$ (for readouts of cells in A'^*) and $r^{B'*}$ (for readouts of cells in B'^*).

Algorithm. For each vector b in the set of optimal pseudo-perturbations, relating cells c_1 (in A') and c_2 (in B'):

1. Compute a set of *redundant cells* for each class. This involves identifying cells in class A with an identical binarized vector b, denoted as set R_b^A, and likewise for class B denoted as R_b^B. Both sets, R_b^A and R_b^B, include cells c_1 and c_2 respectively.
2. Iterate across all pairs of cells in $R_b^A \times R_b^B$, and calculate the difference of readout genes values while keeping the maximal difference.

We retrieve an association of each optimal pseudo-perturbation to a vector of normalized readouts expression that maximizes the difference between the two classes. Additionally, we calculate the *representativity score* for the optimal pseudo-perturbations by considering the number of redundant cells. Let n^A be the number of cells in class A, and let O be the set of Boolean vectors in all optimal pseudo-perturbations for class A. The representativity score S^A for class A is defined as follows:

$$S^A = \frac{\sum_{b \in O} |R_b^A|}{n^A} \times 100. \quad (1)$$

2.4 Implementation and software availability

The complete framework was implemented in an open-source system `scRNA2BoNI` available at: https://github.com/mathieubolteau/scRNA2BoNI. `scRNA2BoNI` uses Answer Set Programming (ASP) [1] as logical modeling and constraint solving paradigm to *identify the maximal number of pseudo-perturbations* and Python for the *maximization of readout difference*. ASP is used to model problems from NP and provides state-of-the-art solvers that propose exact solutions for optimization problems and allow enumeration of all optimal or pseudo-optimal solutions. For our study, we used `clasp` [5]. On a computer cluster comprising 160 CPUs and 1.5 To of RAM, given an association matrix comprising expression of 111 genes for 680 cells, our pipeline allows us to generate 20

pseudo-perturbations in 65h. This corresponds to a pseudo-optimal solution for this problem that is not unique. The ASP program of this algorithm is provided in the supplementary material. The complexity of our program can be analyzed considering two factors that create the search space: (i) the selection of k genes from a total set of G genes, and (ii) the choice of pairs of cells. That is, for each possible selection of k genes, an amount of c associations between cells in classes A and B (where the values of the k genes coincide) has to be tested to discard redundancies within the same class. The maximum value for c is $|A| \times |B|$; which represents associating all cells in both classes. `clasp` performs backjump and conflict-driven learning, optimizing the search space; thus, our estimate measures a worse case. The estimated complexity for the worst-case (see Equation 2) implies that our algorithm is exponential on the number of considered genes and cells from our scRNAseq dataset.

$$\mathcal{O}(\binom{|G|}{k} \times 2^{|A|\times|B|}) \tag{2}$$

3 Results

Our data and results are available as supplementary material at: https://github.com/mathieubolteau/scRNA2BoNI/tree/master/ISBRA_2023_Supp.

3.1 Pseudo-perturbations identification - different size benchmarks

We tested our algorithm on 4 toy datasets (see specifications in Table 1, datasets $A - D$). We also applied our program on 2 entire datasets: phosphoproteomics data, measuring averaged cell population protein expression (dataset P) from [2] and scRNAseq data (dataset SC) from [9]. Our results are shown in Table 1. We can see that using dataset B we identified 5 optimal pseudo-perturbations with identical input and intermediate genes expression for both classes. These 5 different Boolean vectors of pseudo-perturbations represent the expression behavior of 83% (resp. 100%) of the cells in class *early TE* (resp. *medium TE*) for the $k = 5$ selected genes (see Equation 1). On datasets $A - B$, we found an optimal solution, whereas on datasets $C - SC$, suboptimal ones. Our results enable us to advise potential users on expected computation times based on their dataset sizes. For datasets P and SC, we found up to 23 and 20 pseudo-perturbations, respectively. The representativity of selected patients in the phosphoproteomics data (21% and 45%) is vastly lower than the representativity of selected cells in the scRNAseq case study (75% and 89%), suggesting more redundancies in scRNAseq data. Our method is thus applicable for selecting optimal pseudo-perturbations from scRNAseq data.

3.2 Discrimination of the medium and late trophectoderm stages

PKN reconstruction We used 438 transcription factor (TF) genes involved in human embryonic development as input for pyBRAvo to build the PKN (see

Table 1: Maximizing number of pseudo-perturbations applied to 6 case studies.

Dataset	Source	Classes (C1;C2)	m	Cells or patients (C1;C2)	k	Execution time[1]	Different Boolean vectors	Representative score S[2] (C1;C2)
A	artificial	C1 ; C2	10	10 (5;5)	3	0.105s	3	50;60
B	subset of single-cell data	early TE ; medium TE	30	24 (12;12)	5	11.379s	5	83;100
C	subset of single-cell data	early TE ; medium TE	100	50 (25;25)	10	5h*	11	76;80
D	subset of single-cell data	early TE ; medium TE	120	200 (100;100)	15	5h*	18	40;37
P	phosphoproteomics data from [2]	CR ; PR	79	191 (136;55)	10	96h*	23	21;45
SC	single-cell data	medium TE ; late TE	111	680 (348;332)	10	65h*	20	75;89

m refers to the number of input and intermediate genes or proteins. CR = Complete Remission ; PR = Primary Resistant (cf. to [2]). [1] Tests were performed on a computer cluster comprising 160 CPUs and 1.5 To of RAM. [2] see Equation 1. * Execution time corresponds to the fixed timeout.

supplementary data for further details). The PKN is composed of 327 nodes and 475 edges, with only 28 of the 438 initial TFs found in Pathway Commons [12]. We then reduced the network to 191 nodes (84 input, 27 intermediate, 14 readout genes, and 66 complexes) and 285 edges, limited to genes measured in scRNAseq data and complexes linked to these genes (see supplementary material).

Experimental design construction We generated pseudo-perturbations for the experimental design using the method described in Section 2.3, which employed the set of input and intermediate genes from the reduced PKN, comprising 111 genes. Our analysis focused on the expression of these genes across 680 cells, which were identified to be in medium and late TE developmental stages (see Table 1, dataset SC).

We tested different values of k, the number of selected genes, similar to those used in [13]. We observed the number of pseudo-perturbations generated after 30 hours of calculation on a computer cluster and computed the representativity score for each k value. Based on our results, $k = 10$ was the best compromise between a high number of pseudo-perturbations and a high representativity score (see Fig. 1A). This value was also used in [2], supporting our decision.

Our method produced 20 pseudo-perturbation Boolean vectors, which paired medium and late TE cells to maximize the expression value difference of 14 readout genes. In Fig. 1C, we present the experimental design composed of 24 genes: 7 inputs genes (in green), 3 intermediate genes (in red), and 14 readouts (in blue). Each row represents a pseudo-perturbation (on the left, ordered from most to least representative) and its readout observations. Note that each vector is unique. We observe some readout genes with minimal variations (mean of expression difference between both stages less than 0.06), *e.g. DEC1* or *SOD1*, and some readout genes where a significant variation (mean of expression difference

between both stages greater than 0.30) is observed, *e.g. CEBPB*, *CEBPD* or *GSR*. These last also appear in the learned BNs (see Fig. 1B).

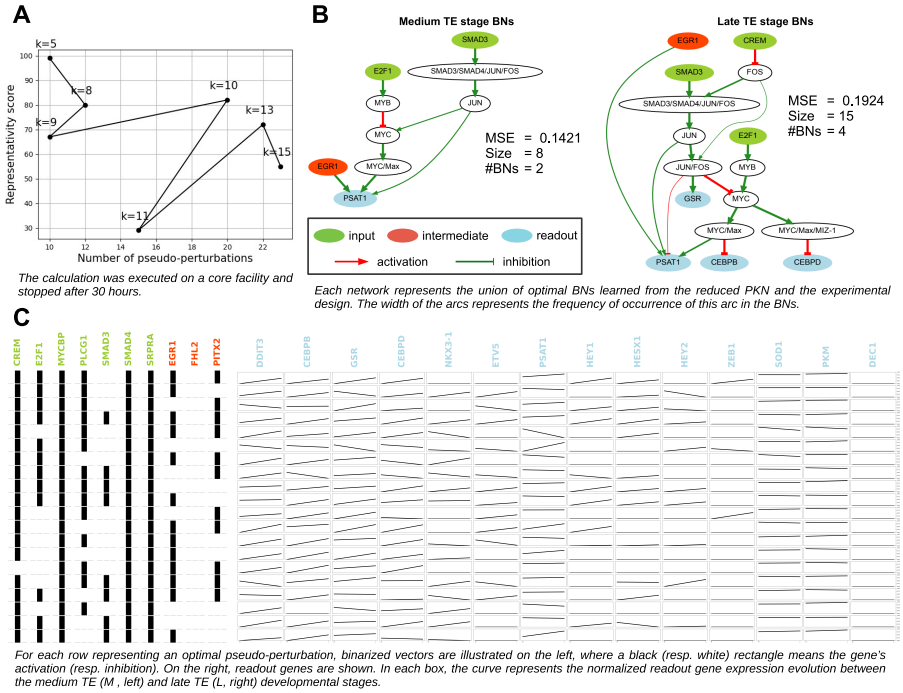

Fig. 1: Medium and late TE discrimination findings. **A.** Impact of k on the number of different pseudo-perturbations and their representativity in the dataset. **B.** Inferred BNs. **C.** Visualization of the experimental design.

BN inference We used the generated experimental design combined with the reduced PKN to infer BNs specific to medium and late TE using the Caspo software. Caspo proposes BNs that match the PKN topology and have an optimal (minimal) mean square error (MSE) between the Boolean prediction of readout nodes (given the Boolean input states) and their experimental measurement. The Caspo used parameters are presented in the supplementary material.

Fig. 1B illustrates the union of learned BNs for the medium and late TE developmental stages, respectively, which are compatible with the fixed fitness value. The size of the learned BNs is equal to 8 for medium TE and 15 for late TE. The optimal MSE for the learned BNs equals 0.1421 and 0.1924, respectively. The medium TE family has 2 BNs, while the late TE one has 4. The execution time for both classes is comparable. These two families of BNs exhibit distinct differences

in their gene behaviors within cell types. Interestingly, the late TE BNs connect more input and readout genes than the medium TE BNs. Both classes of BNs share two input genes, *SMAD3* and *E2F1*, as well as one intermediate gene, *EGR1*, while only one common readout, *PSAT1*, is present. Notably, most of the interactions (without considering their sign) of the medium TE BNs are included in the late TE BNs. While both medium and late TE BNs propose different regulatory mechanisms for PSAT1, the medium TE BNs suggest an activation path from SMAD3. In contrast, the late TE BNs propose two inhibition paths from the same input. Likewise, an inhibition path from *E2F1* to *PSAT1* is proposed in medium TE BNs, while an activation path between these genes is proposed for late TE BNs. This path is, however, subject to the presence or absence of SMAD3. Seemingly the *PSAT1* readout was measured differently in the same pseudo-perturbation configuration involving genes *E2F1* and *PSAT1*. Late TE BNs exhibit supplementary readout genes, namely *GSR*, *CEBPB*, and *CEBPD*, indicating that the readout measurements matched the late TE BNs prediction, given the selected pseudo-perturbation Boolean vectors. However, medium TE BNs could not predict the observed measurements with minimal error on these three genes. Consequently, late TE regulatory mechanisms appear more complex than medium TE ones.

4 Discussion and conclusion

In this paper, we propose an original framework to compute families of Boolean networks compatible with scRNAseq data and prior regulatory knowledge. Our method generates Boolean networks comparing two different conditions. We applied the implemented framework to human embryo development to study the difference between cell behavior at a medium and late TE developmental stage. Despite the lack of in vitro perturbation data and the sparsity of single-cell datasets, our method yields meaningful results.

As significant results, we developed an algorithm to obtain pseudo-perturbations from scRNAseq data demonstrating scalability and efficiency through benchmarking with datasets of varying sizes. The worst-case search complexity for the real case study was of $\binom{111}{10} \times 2^{348\times332} = 3.26 \times 10^{34793}$, and our partial results were generated in 65h. We prove that our algorithm allows for more diverse pseudo-perturbation sets than the state-of-the-art method [2] (see supplementary material), which studied cell population-averaged measurements. We can simulate real perturbations by identifying pseudo-perturbations and proposing more precise (such as Boolean) computational models. Our method identified 20 pairs of cells with Boolean expressions coinciding with selected genes, representing of 75% and 89% of the complete set of cells in medium and late TE developmental stages, respectively.

Using diverse pseudo-perturbations sets, we generate families of Boolean networks to distinguish medium and late TE developmental stages in human embryonic development. The BNs propose Boolean functions derived from the Pathway Commons database to model gene regulation mechanisms. Late TE cells exhibit

a more complex BN structure (size 15 vs. 8) than medium TE cells. These findings are consistent with the fact that late TE requires a gain of biological function to help the embryo implant in the endometrium. Differently, from methods that propose a single computational model of averaged cells, our method includes a subset of 20 cells for each stage and learns optimal families of BNs representing the diversity of expression mechanisms within this cell subset for each stage.

Acknowledgments This work was partially supported by funds from the Agence Nationale de la Recherche [ANR-20-THIA-0011 to M.B., ANR-20-CE17-0007 to L.D. and M.B.]. We are most grateful to the Bioinformatics Core Facility of Nantes BiRD, member of Biogenouest, Institut Français de Bioinformatique (IFB) (ANR-11-INBS-0013) for the use of its resources and for its technical support. Some of the experimentations were performed using HPC resources from the GLiCID computing center.

Appendix

Boolean network (BN) A Boolean network B, of dimension n is defined as $B = (N, F)$ where: $N = \{v_1, \ldots, v_n\}$ is a finite set of nodes (variables or genes) and $F = \{f_1, \ldots, f_n\}$ is a set of Boolean functions $f_i : \mathbb{B}^n \to \mathbb{B}$, with $\mathbb{B} = \{0, 1\}$, describing the evolution of variable v_i.

Influence graph (IG) An IG is denoted by $G = (V, E, \sigma)$ with $V = \{v_1, \ldots, v_n\}$ the set of nodes, $E \subseteq V \times V$ the set of directed edges, and $\sigma \subseteq E \times \{+1, -1\}$ the signs of the edges.

In the context of gene regulation, $j \to i$ means that the change of j in time influences the level of i. Edges $j \to i$ are labeled with a sign, where $+1$ (resp. -1) indicates that j tends to increase (decrease) the level of i. The IG derived from regulatory knowledge bases, is called a *Prior-Knowledge Network* (PKN). The PKN serves as the initial base for generating multiple BNs that adhere to its topology. So that each node in the PKN corresponds to a gene and has an on/off state determined by the Boolean function defined by the BN. Different BNs can have the same IG, while a BN can only be assimilated to a single IG.

Within the PKN, we identify three types of genes. An *input gene*, which is a gene without any predecessor; an *intermediate gene*, with predecessor(s) and successor(s); and a *readout gene*, without any successor. Input and intermediate genes refer to the part of the PKN that can be stimulated (externally or internally), they can also be referred to as system *entries*. While readouts are the part of the system that can be observed, they can be referred to as the system *output*.

Pseudo-perturbations Usually perturbation data is required to discover Boolean mechanisms within a system. This data comes in the form of *on/off* values of

entries associated with output values. However, in the human embryonic development context, perturbing the system is not feasible for obvious reasons. Therefore, we introduce the notion of *pseudo-perturbations*, which refers to artificial perturbations derived from the (unperturbed) gene expression observations.

References

1. Baral, C.: Knowledge Representation, Reasoning, and Declarative Problem Solving. Cambridge University Press, New York, NY, USA (2003)
2. Chebouba, L., Miannay, B., Boughaci, D., Guziolowski, C.: Discriminate the response of acute myeloid leukemia patients to treatment by using proteomics data and answer set programming. BMC Bioinformatics **19**(2), 15–26 (2018)
3. Chevalier, S., Noël, V., Calzone, L., Zinovyev, A., Paulevé, L., Paulevé: Synthesis and simulation of ensembles of boolean networks for cell fate decision pp. 193–209 (2020)
4. Dunn, S.J., Li, M.A., Carbognin, E., Smith, A., Martello, G.: A common molecular logic determines embryonic stem cell self-renewal and reprogramming. The EMBO Journal **38**(1), e100003 (Jan 2019). https://doi.org/10.15252/embj.2018100003
5. Gebser, M., Kaufmann, B., Schaub, T.: Conflict-driven answer set solving: From theory to practice. Artificial Intelligence **187-188**, 52–89 (Aug 2012)
6. Lefebvre, M., Gaignard, A., Folschette, M., Bourdon, J., Guziolowski, C.: Large-scale regulatory and signaling network assembly through linked open data. Database **2021** (2021). https://doi.org/10.1093/database/baaa113
7. Luecken, M.D., Theis, F.J.: Current best practices in single-cell RNA-seq analysis: a tutorial. Molecular Systems Biology **15**(6), e8746 (Jun 2019)
8. McInnes, L., Healy, J., Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction (Sep 2020)
9. Meistermann, D., Bruneau, A., Loubersac, S., Reignier, A., Firmin, J., François-Campion, V., Kilens, S., Lelièvre, Y., Lammers, J., Feyeux, M., Hulin, P., Nedellec, S., Bretin, B., Castel, G., Allègre, N., Covin, S., Bihouée, A., Soumillon, M., Mikkelsen, T., Barrière, P., Chazaud, C., Chappell, J., Pasque, V., Bourdon, J., Fréour, T., David, L.: Integrated pseudotime analysis of human pre-implantation embryo single-cell transcriptomes reveals the dynamics of lineage specification. Cell Stem Cell **28**(9), 1625–1640.e6 (Sep 2021)
10. Petropoulos, S., Edsgärd, D., Reinius, B., Deng, Q., Panula, S.P., Codeluppi, S., Plaza Reyes, A., Linnarsson, S., Sandberg, R., Lanner, F.: Single-Cell RNA-Seq Reveals Lineage and X Chromosome Dynamics in Human Preimplantation Embryos. Cell **165**(4), 1012–1026 (May 2016). https://doi.org/10.1016/j.cell.2016.03.023
11. Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H.A., Trapnell, C.: Reversed graph embedding resolves complex single-cell trajectories. Nature Methods 2017 14:10 **14**(10), 979–982 (2017). https://doi.org/10.1038/nmeth.4402
12. Rodchenkov, I., Babur, O., Luna, A., Aksoy, B.A., Wong, J.V., Fong, D., Franz, M., Siper, M.C., Cheung, M., Wrana, M., Mistry, H., Mosier, L., Dlin, J., Wen, Q., O'Callaghan, C., Li, W., Elder, G., Smith, P.T., Dallago, C., Cerami, E., Gross, B., Dogrusoz, U., Demir, E., Bader, G.D., Sander, C.: Pathway Commons 2019 Update: integration, analysis and exploration of pathway data. Nucleic Acids Research **48**(D1), D489–D497 (10 2019). https://doi.org/10.1093/nar/gkz946
13. Videla, S., Saez-Rodriguez, J., Guziolowski, C., Siegel, A.: caspo: a toolbox for automated reasoning on the response of logical signaling networks families. Bioinformatics **33**(6), 947–950 (Mar 2017). https://doi.org/10.1093/bioinformatics/btw738