A Parallel Approach for Accelerated Parameter Identification of Gene Regulatory Networks

Tariq Saeed and Jamil Ahmad

Research Centre for Modeling and Simulation (RCMS),
National University of Sciences and Technology (NUST), Pakistan
{tariq,jamil.ahmad}@rcms.nust.edu.pk
http://www.rcms.nust.edu.pk

Abstract. Model Checking is one of the formal verification methods which can be used to infer parameters of a Gene Regulatory Network (GRN) using discrete formalism of René Thomas. However, the sequential approach for identification of these logical parameters is computationally intensive and takes lot of processing time, depending on number of genes and range of their expression levels in a network. In this paper, we present an efficient approach for this problem, based on parallel computing. We partition the parameter space into subsets and assign them to processing elements on a distributed memory parallel computer. The presented approach is implemented by using OpenMPI and existing tool SMBioNet. The experimental results indicate that the approach is scalable and achieves 7X speed-up, for a relatively small GRN, comprising of five genes.

Keywords: Parallel SMBioNet, Gene Regulatory Network, Discrete Modeling

1 Introduction

Living cells are complex power houses of human machinery, where the activities are organized as network of interacting entities such as Genes, mRNA and their products. Understanding the structural and behavioral orientation of cellular and sub-cellular interactions is fundamental step to uncover disease mechanics and realization of personalized medicine [4]. As the complexity of these networks increase, efficient computational methods are required not only to infer network information from gene expression data, but also to gain insight into regulatory mechanisms by finding answers to biological questions [6]. This growing complexity of GRNs also requires correlation mechanisms between experimental data, theory and computational methods [3]. Once a suitable framework is selected, and the system under investigation has been modeled, identification of parameters, coherent with biological knowledge, is a key problem known as Parameter Estimation or Network Inference [5]. These parameters are not a priori known, and have to be reverse engineered and compared with biological knowledge. The problem of parameter estimation is extremely important in systems biology and directly effects the accuracy of results. Several computational methods have been used for modeling and analysis of regulations at gene level. Model Checking [2] is one of the widely used formal verification methods in this area, due to its automated approach and predictive ability to answer questions.

In this paper, we present a parallel approach for logical parameter identification of GRNs, and report experimental results on 8 nodes of High Performance Computing (HPC) cluster. Our approach is based on well known discrete modeling framework of René Thomas [1]. In this formalism, a GRN is represented as labeled directed graph and its dynamics are derived from a set of logical parameters, which are unknown and can be calculated using model checking. We divide the set of parameter combinations among processing elements by using a block-wise decomposition scheme. Each processing element retains only those parameters which are verified by model checker. We have performed a prototype implementation of our approach by modifying an existing implementation of SMBioNet [12]. The experimental results indicate that presented approach is scalable and achieves almost linear speed-up.

The rest of the paper is organized as follows; in Sec. 2, we provide a brief background of René Thomas formalism and recall some associated definitions, followed by sequential algorithm for paramter identification using model checking in Sec 2.4. In Sec. 3, we discuss our parallel approach, its implementation, and performance evaluation. Finally, Conclusion and Future work is presented in Sec. 4

2 Parameter Identification through Model Checking

2.1 Discrete Modeling Framework

Numerous approaches have been proposed for modeling regulatory networks [7]. Continuous modeling approaches using ordinary and partial differential equations have been applied but analytical solutions of these methods are difficult to solve due to non-linearity of gene regulations. Numerical solution can be computed but their accuracy depends on the value of logical parameters, which are not experimentally measurable. These limitations led Thomas to propose a simplification of biological models in the from of a discrete modeling framework [1]. The main advantage of using discrete model is that the number of logical parameters are finite. we provide an introduction of this framework and recall some associated definitions. For more detailed review of semantics of this framework, we refer to [13]

Definition 1. *A* Gene Regulatory Network (GRN) *is a labeled directed graph* $G = (V, E)$*, where* $V = \{v_1, ..., v_n\}$ *is set of nodes in which each* $v_i \in V$ *represents a biological entity in the network and* $E = \{e_1, ..., e_n\}$ *is set of edges in which each* $e_{v_x,v_y} \in E$ *represents interaction between a pair of biological entities* v_x *and* v_y*. Each edge* $v_x \to v_y$ *is labeled by a pair* $(t_{x,y}, \alpha_{x,y})$*, where* $(t_{x,y})$*, called threshold, is a positive integer and* $(\alpha_{x,y}) \in \{+, -\}$ *is the sign of interaction (+ for activation and - for inhibition). Each node* $v_i \in V$ *is provided with a limit* ℓ_{v_i}*, which is equal to the out-degree of* v_i*, and* $\ell_{v_i} = 1$*, if out-degree is zero.*

In a directed graph $G = (V, E)$, we denote set of predecessors and successors of a variable $v_i \in V$ as $G^-_{v_i}$ and $G^+_{v_i}$ respectively. We represent a qualitative state like a vector containing individual expression level of each variable in GRN.

Definition 2. *Let $G = (V, E)$ be a GRN. A* State of GRN *is n-tuple $S = \{s_{v_1}, .., s_{v_n}\}$, where n is the number of variables in the network and s_{v_i} is the abstract expression level of v_i with $s_{v_i} \in \mathbb{N}$ and $s_{v_i} \leq \ell_{v_i}$.*

The total number of states in a GRN G with n variables are given as $\prod_{i=1}^{n}(\ell_{v_i} + 1) \forall v \in V$, where ℓ_{v_i} denotes the maximum expression level of variable v_i in the network. The number of effective regulators of a variable v_i at expression level s_{v_i}, is formally represented by its set of resources.

Definition 3. *Let $G = (V, E)$ be a GRN. The* Set of Resources *ω_{v_y} of a variable $v_y \in V$, at level s_{v_y}, is defined as $\omega_{v_y} = \{v_x \in G^-_{v_y} \mid (s_{v_x} \geq t_{x,y}$ and $\alpha_{x,y} = +)$ or $(s_{v_x} < t_{x,y}$ and $\alpha_{x,y} = -)\}$*

While determining the resource set of a variable, presence of an activator or absence of inhibitor are treated as resource. The set ω_{v_y} contains the inhibitors of v_y with expression levels below the threshold and activators of v_y whose expression levels are greater than or equal to threshold. The target towards which the variable v_y evolves, when its resource set is ω_{v_y}, is given by a set of positive integers $K_{\omega_{v_y}} \leq \ell_{v_y}$. When variable v_y is at a certain expression level $s_{v_y} \in S$, the next level towards which S_{v_y} evolves, is determined by the value of its logical parameters $K_{\omega_{v_y}}$ and there are three possibilities. When $s_{v_y} < K_{\omega_{v_y}}$, the value of s_{v_y} can be incremented by one unit. Conversely, if $s_{v_y} > K_{\omega_{v_y}}$, s_{v_y} can be decremented by one unit. However, if $s_{v_y} = K_{\omega_{v_y}}$, s_{v_y} does not evolve and remains constant. The state graph is generated using the values of these logical parameters. The number of possible parameter combinations can be huge, even for a small network. Given a GRN $G = (V, E)$, with n variables, total number of parameterizations are given as $\prod_{i=1}^{n}(\ell_{v_i} + 1)^{2^{|G^-(v_i)|}}$, where $|G^-(v_i)|$ represents cardinality of the set of regulators of v_i.

2.2 Example 1 (Mucus Production in Pseudomonas aeruginosa)

We apply Thomas' framework on GRN of mucus production system in Pseudomonas aeruginosa [10], an opportunistic pathogen that secretes mucus in lungs effected by cystic fibrosis. It causes respiratory deficiency in the patients and is a major cause of mortality in lungs diseases. The GRN involved in mucus production mainly comprises of two genes; AlgU, which is the main regulator of mucus production, and its inhibitor. This network can be modeled as a labeled directed graph (Figure 1a), in which variable x represents AlgU and variable y represents its inhibitor. The interactions are shown as directed edges. The edge from variable x to variable y with positive sign shows that AlgU favors the production of its inhibitor. Conversely, the edge from variable y to variable x is labeled with negative sign, which shows that y has negative influence on the production of

AlgU. A Self loop on x represents the fact that presence of AlgU also favors its own production, when its level reaches 2. In order to deduce dynamics, the GRN is translated into a state graph (Figure 1b) by using some parameter values. The number of parameterizations for a given GRN depend on number of genes, their regulatory relationship and range of their expression levels. Different parameterizations of the same network may lead to different state graphs and consequently to unique biological behaviors. In case of Psedomonas, variable x can take three values $\{0,1,2\}$, whereas variable y can take two values $\{0,1\}$. In this case, the total number of states are six. Each variable v has $2^{|(G^-(v_i))|}$ regulators, where $|G^-(v_i)|$ represents the in-degree of v. This calculation leads to 324 parameter combinations for asynchronous dynamics, even for a network that only involves two genes.

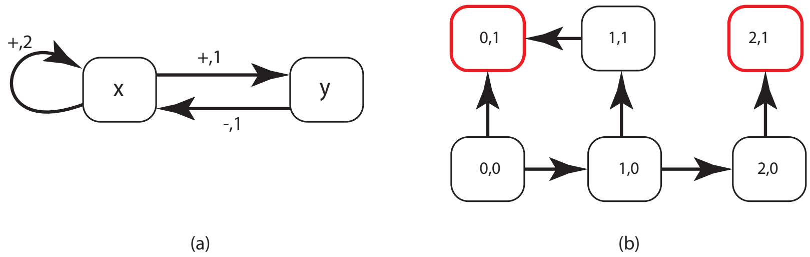

Fig. 1: (a) Simple Interaction Graph of two Genes X and Y, Each interaction is labeled with threshold and nature of influence, + sign shows activation and - shows Inhibition), (b) State Graph shows two stable states $[(0,1),(2,1)]$ with following parameter values; $K_x\{\}=0$, $K_x\{x\}=2$, $K_x\{y\}=2$, $K_x\{x,y\}=2$, $K_y\{\}=1$, $K_y\{\}=1$

2.3 Model Checking

Modeling regulatory networks using Thomas' framework leads to qualitative model (state graph), and graph traversal algorithms can be employed to infer simple behaviors such as deadlocks. However, a formal temporal language is required for encoding complex behaviors. The advantage of expressing these behaviors as temporal logic properties is that the formal methods in computer science can automatically check whether any biological system possesses these properties or not. More importantly, formal methods are used to reverse engineer the values of logical parameters that satisfy these temporal properties [11,15]. Model Checking technique is one of the formal verification method which has been used widely for automated verification of complex systems. The model checking approaches can be differentiated on the basis of how they interpret the notion of time. Linear or branching. Due to branching nature of CTL (**C**omputation **T**ree

Logic), it is naturally suited for non deterministic dynamical systems such as GRNs, where a current state can have more than one successor states. Model checking is not only useful to check biological properties of GRN but also to establish and test new hypothesis and compute associated logical parameters.

In a CTL formula, $\top$ is always taken as true; $\bot$ is always false; $(v_i = n)$ is true iff expression level of variable v_i, in current state, is equal to n. The CTL formula combines a set of connectives: $\neg$(negation), $\wedge$ (logical AND), $\vee$ (logical OR) and $\Rightarrow$ (implication) with temporal operators. The temporal operators are pairs of symbols; the first element of which is A (all paths) or E (at least one path), followed by X (next state), F (any future state) or G (all future states).

Definition 4. *Let $G = (V, E)$ be a GRN. The* CTL Formula *Φ on G is defined as follows;*

- *atomic formulas are $\top$, $\bot$ or any atomic proposition of the form $(v_i = n)$, where v_i is a variable in state graph and $n \in [0, \ell_{v_i}]$.*
- *If ϕ and ψ are atomic formulas, then so are $(\neg\phi)$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \Rightarrow \psi)$, $X\phi$, $EX\phi$, $AG\phi$, $EG\phi$, $EF\phi$, $AF\phi$, $(A\phi \bigcup \psi)$ and $(E\phi \bigcup \psi)$*

2.4 Sequential Algorithm

The sequential algorithm for parameter identification (Figure 2) comprises of three main steps.

- First, the regulatory network is modeled using Thomas' framework as labeled directed graph.
- Secondly, list of all possible parameterizations is generated after application of snoussi constraints [19], and behavioral properties are encoded in CTL format according to definition 4. Based on an exhaustive enumeration, a state graph is generated for each parameterization. The state graph and the CTL property is supplied to model checker. If CTL property is satisfied, the set of parameters are retained as accepted parameters.
- Finally, the set of all models/parameterizations that satisfy the given CTL properties are returned as the output.

SMBioNet [11, 12] is based on Thomas' formalism and implements the sequential approach for parameter identification. It has been used for the analysis of several regulatory networks, including tail resorption in tadpole metamorphosis [12] and immunity control in bacteriophage lambda [15]. The details of GRN and CTL properties are specified in the form of input file and the output contains all models that satisfy these properties. For each parameterization, it invokes NuSMV [8] for verification. On an abstract level, the functionality of SMBioNet can be categorized into model generation phase and verification phase. In model generation, it encodes the details of state graph into NuSMV input format, whereas in verification phase, it calls NuSMV model checker. For a complete GRN $G = (V, E)$ on n variables, the complexity of SMBioNet is given

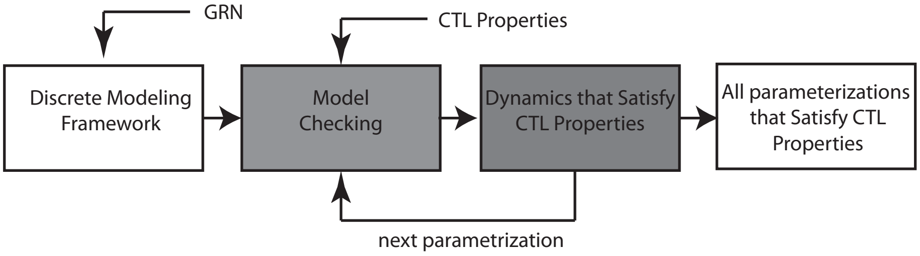
GRN
CTL Properties
Discrete Modeling Framework
Model Checking
Dynamics that Satisfy CTL Properties
All parameterizations that Satisfy CTL Properties
next parametrization

Fig. 2: Sequential Flow of Parameter Identification, Shaded stages are executed in loop until all parameter combinations are verified using model checker. Finally, all parameterizations for which CTL formula is satisfied are retained.

as $(2^n)^{(2^n)}$, due to which it works for small networks, typically $n < 7$ [14]. In order to understand, where most of the processing time is spent in sequential algorithm, we obtain a profiling information by de-linking the model generation and verification phases in SMBioNet and provide a model of MAL-Associated GRN(Sec. 3.2) as input. The results indicate that more than **95%** of total execution time is spent in the verification phase. Therefore, the decomposition of verification phase can lead to significant overall performance enhancement.

3 Parallel Approach

The parallel approach presented here, is based on the concept of Data Decomposition. In classical theory of parallel computing, decomposition of of a problem can be carried out with respect to partitioning of data, tasks or both [18]. A technique that splits large data into subsets, and associates same operation with different chunks of data, is known as Data Decomposition. It is clear from the sequential algorithm that the parameterizations are exponentially large and for each combination, a state graph is generated and verified against CTL properties. Since there is no dependency between the generation of any two state graphs, it is possible to generate multiple state graphs from a list of parameter combinations. These state graphs can be generated in parallel and supplied to a model checking process for the purpose of verification. Keeping in view this embarrassingly-parallel configuration, we employ a simple master-worker model of computation, in which master process is responsible for pre-processing, generation of state graphs and communication with worker processes, whereas parameter identification is carried out by worker processes. We perform domain decomposition on the set of all possible parameters ρ, to distribute the verification tasks among n worker processes $w_1, ..., w_n$. The algorithms for master process is given as Algorithm 1 which starts by initializing three lists: ρ for storing all parameter combinations; $\wp$ to store only those combinations for which CTL formula Φ is satisfied and $\Re$ for storing state graphs generated from each combination in ρ. The procedure *build* (line 4) performs pre-processing on GRN G to generate complete list of all parameters after applying constraints. In a *for* *loop* (line 6-8), the master process generates a state graph for each parameter combination and stores it in the $\Re$.

Algorithm 1: domain decomposition on parameterizations

```
   Data: Gene Regulatory Network: $G$, CTL Formula: $\phi$, Number of Worker
         Processes: $nw$
   Result: Set of all parameterizations satisfying $\phi$
1  $\rho \leftarrow \emptyset$ ;
2  $\wp \leftarrow \emptyset$;
3  $\Re \longleftarrow \emptyset$;
4  $\rho \leftarrow build(G)$ ;
5  $q \leftarrow length(param)$;
6  forall the elements of param do
7      remove an element $j$ from $\rho$;
8      $\Re \leftarrow generate(j)$ ;
9  foreach $w_i \in$ worker processes do
10     $s \leftarrow \lfloor (i * q)/nw \rfloor$ ;
11     $e \leftarrow \lfloor ((i+1) * q/nw \rfloor) - 1$ ;
12     $send(\Re, s, e, w_i)$
13 foreach $w_i \in$ worker processes do
14     recv($\wp, w_i$);
15 return $\wp$;
```

Given m state graphs $\Re = (r_1, .., r_m)$, and nw worker processes, block-wise decomposition partitions $\Re$ onto nw blocks. Each block i, $1 \leq i \leq nw$ contains consecutive elements with indices $(\lfloor i.q/nw \rfloor, \ldots, \lfloor (i+1).q/nw - 1 \rfloor)$, where q is the length of ρ. These indices represent share of each worker process w_i, and is a subset of $\Re$. A block of state graphs is sent to each worker process for verification (line 12). Finally, when the verification is complete by worker processes, and the set of acceptable parameterizations is generated by worker processes, master process collects the results by calling a receive function (line 14). The algorithm for each worker process w_i is given as Algorithm 2, which starts by initializing two lists: namely $R \in \Re$; for storing state graphs received from master process, and *selected*; for storing accepted parameters after verification . In a *for loop* (line 4-8), each worker process enumerates through all state graphs and invokes model checker by calling procedure *check* (line 6) for verification of CTL formula Φ. If the formula is satisfied, the paramter combination is retained in the list *selected.* Finally, the set of accepted parameters are sent to master process (line 9).

Algorithm 2: model checking for parameter identification

```
Data: CTL Formula $\Phi$
Result: subset of all parameterizations satisfying $\Phi$
$R \leftarrow \emptyset$;
$selected \leftarrow \emptyset$;
$R = recv(R, master)$ ;
forall the elements of $R$ do
    remove an element $s$ from $R$;
    $status = check(\text{s}, \Phi)$ ;
    if $(status = verified)$ then
        $selected = getParam(s)$ ;
$send(selected, master)$;
```

3.1 Implementation

We perform a prototype implementation of our presented approach by modifying the existing implementation of SMBioNet to incorporate master-worker model of computation, using OpenMPI [16]. Each process is assigned a rank, which is used for distribution of workload. The model generation phase is implemented as master process which generates all models, from input interaction graph and set of all parameterizations. These models are written as input files for NuSMV model Checker. The worker processes read these input models and invoke NuSMV for verification. Each process calculates its share of workload, based on the total number of parameterizations, its rank, and the number of worker processes. However, verification of models can not be performed until the models are generated. Instead of keeping this dependency as a serial component in our implementation, we enforce a very small delay (few milliseconds) in launching the worker processes so that sufficient number of models are accessible by the time the worker processes are launched. In this way, we are able to execute both model generation and verification concurrently.

3.2 Example 2 (Mal-associated regulatory network)

In order to check the scalability of our implementation, we apply it on MAL-associated regulatory network [9]. In this network (Figure 3), Brutons Tyrosine Kinase (BTK) acts as a positive regulator of MAL. The Inflammatory Cytokines (INCY) generate inflammation and also activate SOCS-1 which inhibits phosphorylated MAL and also degrades NF-kB expression. The parameters associated with BTK: $K_{BTK}\,\{\}$ can take values in the range of $\{0, 1\}$; parameters associated with MAL: $K_{MAL}\,\{\}$, $K_{MAL}\,\{SOCS\}$, $K_{MAL}\,\{BTK\}$ and $K_{MAL}\,\{BTK, SOCS\}$ can take values in the range of $\{0, 1\}$; parameters associated associated with SOCS: $K_{SOCS}\,\{\}$ and $K_{SOCS}\,\{INCY\}$ can take values in the range of $\{0, 1\}$; parameters associated associated with INCY: $K_{INCY}\,\{\}$ and

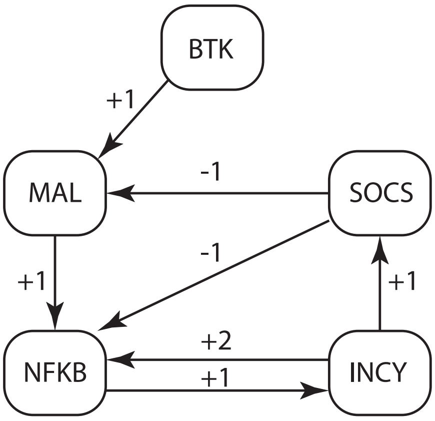

Fig. 3: MAL-associated Regulatory Network adopted form [9] The Numerals (1 and 2) show thresholds of interactions; Arrow labeled with + sign indicate activation while arrow labeled with - sign show inhibition. The direction of arrows indicate the direction of activation/inhibition

$K_{INCY}\{NFKB\}$ can take values in the range of $\{0, 2\}$. Similarly, all parameters associated with NFKB: $K_{NFKB}\{\}$, $K_{NFKB}\{SOCS\}$, $K_{NFKB}\{INCY\}$, $K_{NFKB}\{MAL\}$, $K_{NFKB}\{INCY, SOCS\}$, $K_{NFKB}\{MAL, INCY, SOCS\}$, $K_{NFKB}\{MAL, INCY\}$ and $K_{NFKB}\{MAL, SOCS\}$are in the range of $\{0, 1\}$. This leads to a total combinations of $2^2 * 2^4 * 2^8 * 2^2 * 3^2 = 589824$ parameterizations. After applying observability and snoussi constraints, the parameters are reduced to 4320.

By using definition 4, we encode stable states reported in [9] as CTL formulas (Figure 4) in SMBioNet input file. The first three properties, denoted as $\Phi(ini)$, $\Phi(pat)$ and $\Phi(clr)$ represent initial, pathogenic and reset state respectively, in state graph of Mal-associated regulatory network. The property $\Phi(\alpha) : \Phi(ini) \rightarrow EF(AG(pat))$ shows a particular behavior of the system (state graph) that, from initial state, pathogenic state is reachable in all paths. When this property is verified using SMBioNet, only those models are selected which satisfy $\Phi(\alpha)$. Similarly, $\Phi(\beta) : \Phi(ini) \rightarrow EF(AG(clr))$ represents that systems is able to reach a particular state where the expression level of all the variables is zero. The CTL formula $\Phi = \Phi(\alpha) \wedge \Phi(\beta)$ is only true when both these behaviors are present in the system. Consequently, all models that satisfy $\Phi(MAL)$ are retained as accepted set of parametrizations.

We encode CTL formula (Figure 4) in SMBioNet input file and execute using our modified parallel algorithm on 8 nodes of HPC cluster connected with Infiniband interconnect. Each node has a dual quad-core processor with 24GB of memory. The model is executed with 9 processes. The master process is designated as root process with rank 0, and the ranks of worker processes start from 1 to 8. The running time of the experiments is presented in Table 1. These ex-

```
φ(ini)=  (BTK=1&MAL=0&NFKB=0&SOCS=0&INCY=0)        (1)
φ(pat)=  (BTK=0&MAL=0&NFKB=1&SOCS=1&INCY=2)        (2)
φ(clr)=  (BTK=0&MAL=0&NFKB=0&SOCS=0&INCY=0)        (3)
φ(α):    φ(ini)-> EF(AG(pat))                      (4)
φ(β):    φ(ini)-> EF(AG(clr))                      (6)
φ(MAL)=  φ(α)^φ(β)                                 (7)
```

Fig. 4: CTL formula to determine values of logical parameters in MAL-Associated Regulatory Network, for two stable states reported in [9]

perimental results show that the approach is scalable, even for a relatively small model, comprising of 5 genes. With increase in size of the network, granularity will also increase and the approach is likely to work for large scale models.

Process Count	Running Time (Sec.)	Process Count	Running Time (Sec.)
1	48.12	5	10.12
2	24.72	6	9.16
3	17.85	7	8.02
4	12.85	8	7.12

Table 1: Running Time of parallel algorithm on 8 processors

4 Conclusion and Future Work

In this paper, we presented a parallel approach to improve the efficiency of parameter identification in GRN. Our main contribution is the parallelization of sequential approach for parameter identification in GRNs by decomposition of parameter space into subsets. The presented approach can be used for analysis of large scale regulatory networks under parameter uncertainty. The presented approach is part of our on-going research work for the development of efficient parallel model checking techniques for verification of large scale GRNs, based on René Thomas discrete formalism. In future, we plan to test the approach for large scale models on HPC platform with more number of nodes, Moreover, we plan to use light weight threads in combination with MPI processes for cluster of multicore machines. In the current work, we have implemented our approach using CTL based model checker. Recent progress in the area of LTL-based model checking has demonstrated huge performance gains and our presented approach can be tested with LTL based model checking.

References

1. Thomas, René, Richard d'Ari: Biological feedback. CRC press, (1990)

2. Clarke, Edmund M., Orna Grumberg, Doron A. Peled: Model checking. MIT press, (1999)
3. Albert, Réka: Boolean Modeling of Genetic Regulatory Networks. Complex Networks, pp. 459-481. Springer Berlin Heidelberg (2004)
4. Barabsi, Albert-Lszl, Zoltan N. Oltvai: Network biology: understanding the cell's functional organization. Nature Reviews Genetics (2004)
5. Lillacci, Gabriele, Mustafa Khammash: Parameter estimation and model selection in computational biology. PLoS Computational Biology (2010)
6. Lee, Wei-Po, Wen-Shyong Tzou: Computational methods for discovering gene networks from expression data. Briefings in Bioinformatics (2009)
7. De Jong, Hidde: Modeling and simulation of genetic regulatory systems: a literature review. Journal of computational biology (2002)
8. Cimatti, Alessandro, Edmund Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, Armando Tacchella: Nusmv 2: An opensource tool for symbolic model checking. In: Computer Aided Verification, pp. 359-364. Springer Berlin Heidelberg (2002)
9. Ahmad, Jamil, Umar Niazi, Sajid Mansoor, Umair Siddique, Jaclyn Bibby: Formal Modeling and Analysis of the MAL-Associated Biological Regulatory Network: Insight into Cerebral Malaria. PloS ONE (2012)
10. Jamil Ahmad, Gilles Bernot, Jean-Paul Comet, Didier Lime, Olivier Roux: Hybrid Modelling and Dynamical Analysis of Gene Regulatory Networks with Delays. Complexus (2006)
11. Bernot, Gilles, Jean-Paul Comet, Adrien Richard, Janine Guespin: Application of formal methods to biological regulatory networks: extending Thomas asynchronous logical approach with temporal logic. Journal of theoretical biology (2004)
12. Khalis, Zohra, Jean-Paul Comet, Adrien Richard, Gilles Bernot. The SMBioNet method for discovering models of gene regulatory networks. Genes, Genomes and Genomics (2009)
13. Bernot, Gilles et al.: Semantics of biological regulatory networks. Electronic Notes in Theoretical Computer Science (2007)
14. Richard, Adrien, et al.: Boolean models of biosurfactants production in Pseudomonas fluorescens. PloS one (2012)
15. Richard, Adrien, Jean-Paul Comet, Gilles Bernot: Formal methods for modeling biological regulatory networks. In: Modern Formal Methods and Applications, pp. 83-122. Springer Netherlands (2006)
16. Gabriel, Edgar, Graham E. Fagg, George Bosilca, Thara Angskun, Jack J. Dongarra, Jeffrey M. Squyres, Vishal Sahay et al.: Open MPI: Goals, concept, and design of a next generation MPI implementation. In: Recent Advances in Parallel Virtual Machine and Message Passing Interface, pp. 97-104. Springer Berlin Heidelberg (2004)
17. Filopon, Didier, et al.: Epigenetic acquisition of inducibility of type III cytotoxicity in P. aeruginosa. BMC bioinformatics (2006)
18. Kumar, V., Grama, A., Gupta, A., & Karypis, G.: *Introduction to parallel computing* (1994)
19. Snoussi, El Houssine, and Réne Thomas: Logical identification of all steady states: the concept of feedback loop characteristic states. Bulletin of Mathematical Biology (1993)