



GRNPar:

Parallel Inference of Gene Regulatory Networks Using Boolean Network Models and Mutual Information

William Das & Anushka Gupta



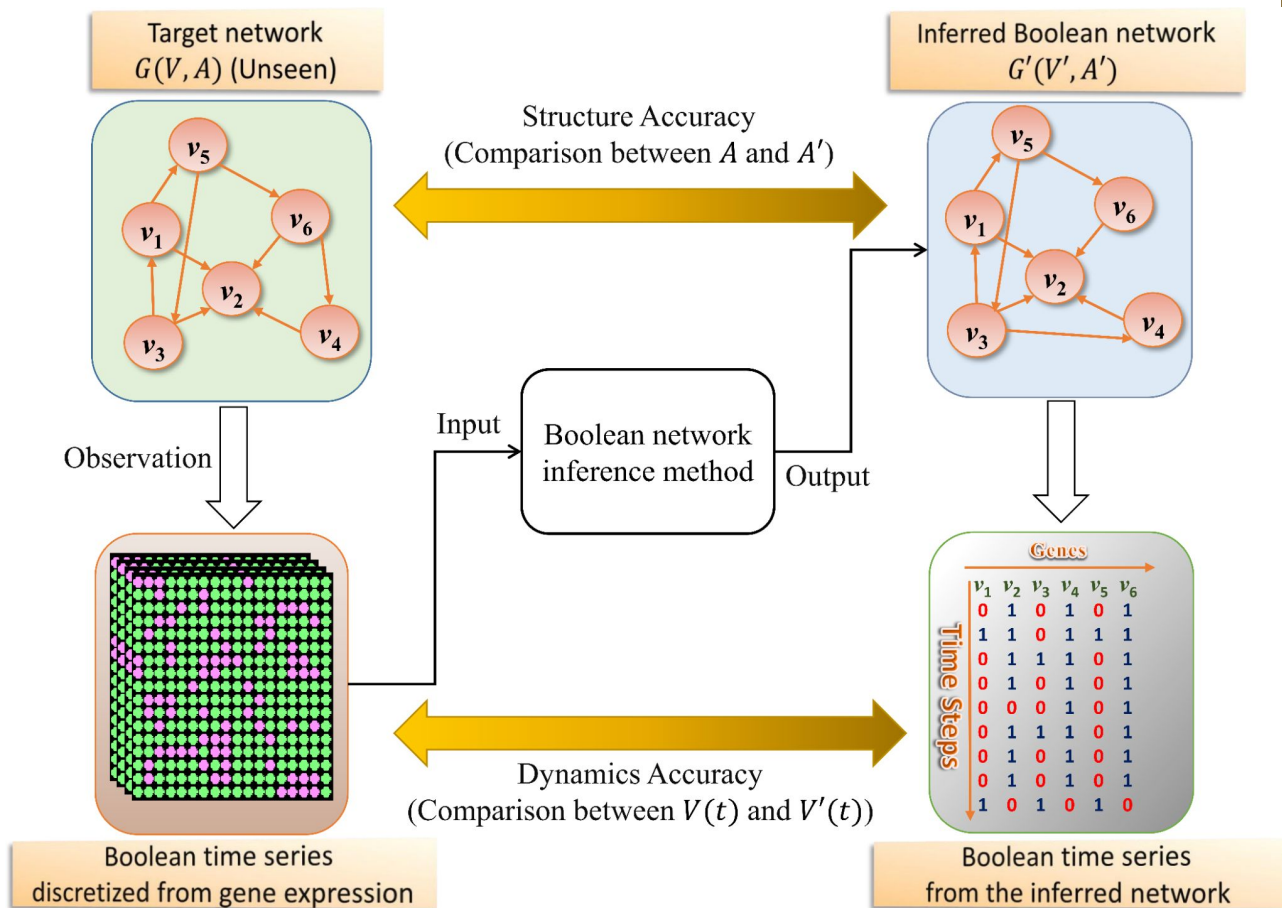
Problem Statement

Input:

Discretized **boolean** time-series gene-expression data

Output:

Boolean Gene Regulatory Network (information theoretic approach with boolean networks)



GRNPar Pipeline

Infer Gene Network Dependencies

Generate Gene-wise Boolean Expressions

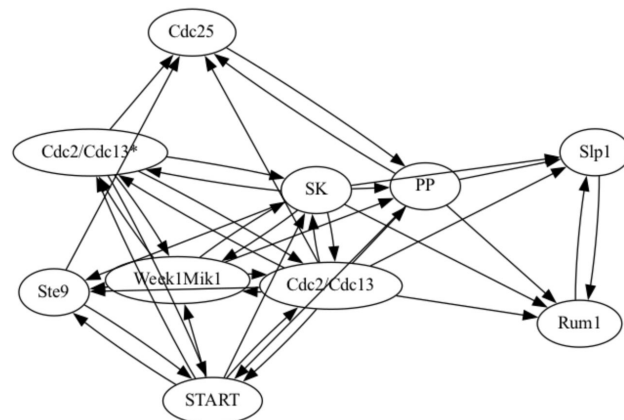
Graph Boolean Network

```
{  
  Slp1: {  
    (Rum1 -> Slp1),  
    (PP -> Slp1),  
    (Cdc2/Cdc13 -> Slp1)},  
  Cdc2/Cdc13*: {  
    (START -> Cdc2/Cdc13*),  
    (Cdc2/Cdc13 -> Cdc2/Cdc13*),  
    (SK -> Cdc2/Cdc13*)},  
  Week1Mik1: {  
    (START -> Week1Mik1),  
    (Cdc2/Cdc13 -> Week1Mik1),  
    (SK -> Week1Mik1)},  
}
```

Slp1 = ((Rum1 & PP) & Cdc2/Cdc13).
Gene-wise dynamics consistency: **0.67**

Cdc2/Cdc13* = ((START & Cdc2/Cdc13) & SK). Gene-wise dynamics consistency: **0.78**

Week1Mik1 = ((START & Cdc2/Cdc13*) & SK). Gene-wise dynamics consistency: **0.78**



Fission yeast cell cycle gene regulatory boolean network (real-world data).

Sequential Method: Mutual Inference Approach

Mutual Inference Algorithm (MIFS)

1. Input: Nodes V , k (# of input nodes)
2. For each node $v_o \in \text{Input nodes } V$
 - a. For each node $w \in V \setminus v_o$
 - i. Determine mutual information (entropy) between v & w
 - ii. Save node if mutual information is amongst highest k

$$E(v, v') = \frac{\sum_{t=2}^T I(v(t) = v'(t))}{T - 1}$$

Gene-wise dynamics consistency

Generate Gene-Wise Boolean Expressions

1. For each input node $v \in k \text{ Input nodes } K$
 - a. Generate combinations of boolean expressions (AND/OR operators) using k input nodes, sorted in descending order based on mutual information
 - b. For each possible expression, determine gene-wise dynamics consistency for each target node:
 - i. For each target node at time $t + 1$, test the boolean expression on input nodes at time t
-> capture results across entire time-series and calculate **proportion of correct predictions**
 - ii. Choose the boolean expression with **highest criterion** for target node

Parallel Method: Mutual Inference Approach

Mutual Inference Algorithm (MIFS)

1. Input: Nodes V , k (# of input nodes)
2. For each node $v_o \in \text{Input nodes } V$ `concat $ parMap rdeepseq`
 - a. For each node $w \in V \setminus v_o$ `map `using` parBuffer 3 rdeepseq`
 - i. Determine mutual information (entropy) between v & w
 - ii. Save node if mutual information is amongst highest k

$$E(v, v') = \frac{\sum_{t=2}^T I(v(t) = v'(t))}{T - 1}$$

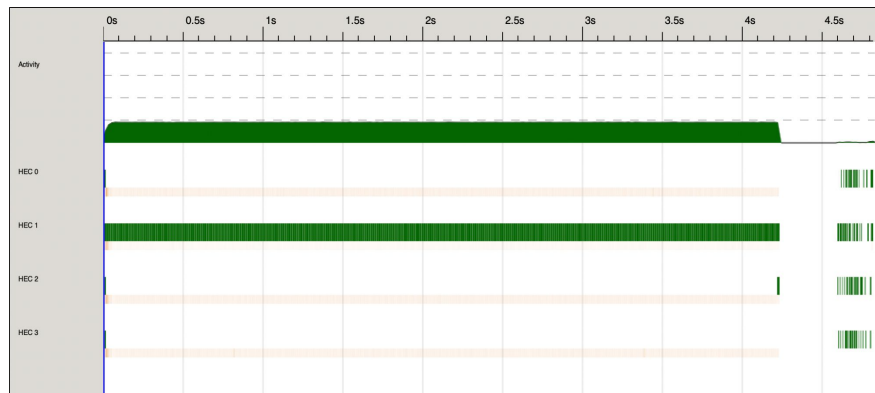
**Gene-wise dynamics
consistency**

Generate Gene-Wise Boolean Expressions

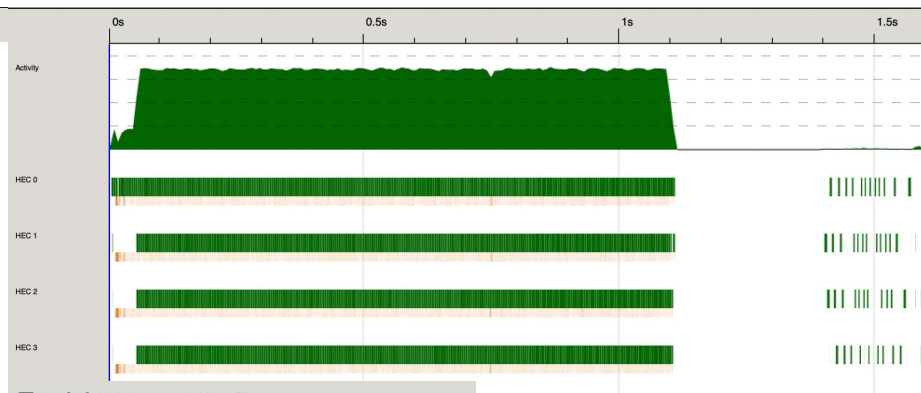
1. For each input node $v \in k$ Input nodes K
 - a. Generate combinations of boolean expressions (AND/OR operators) using k input nodes, sorted in descending order based on mutual information
 - b. For each possible expression, determine gene-wise dynamics consistency for each target node:
 - i. For each target node at time $t + 1$, evaluate the boolean expression (`parMap rdeepseq`) on input nodes at time t -> capture results across entire time-series and calculate **proportion of correct predictions** `map `using` parBuffer 50 rdeepseq`
 - ii. Choose the boolean expression with **highest criterion** for target node

Sequential vs Parallel Implementation

Randomly Generated Data: 100 Nodes over 300 time steps (4 cores)



Total time: 4.818s
Mutator time: 4.656s
GC time: 162.323ms
Productivity: 96.6% of mutator vs total



Total time: 1.654s
Mutator time: 1.564s
GC time: 89.631ms
Productivity: 94.6% of mutator vs total

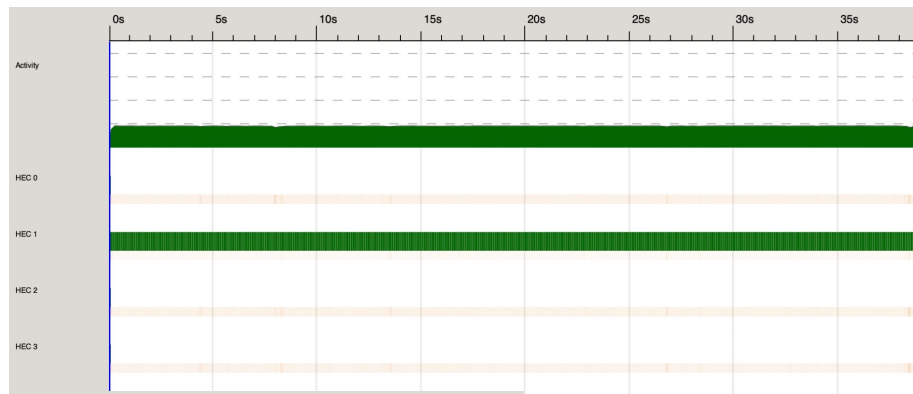
2.91x speedup

2.97x reduction in Mutator Time

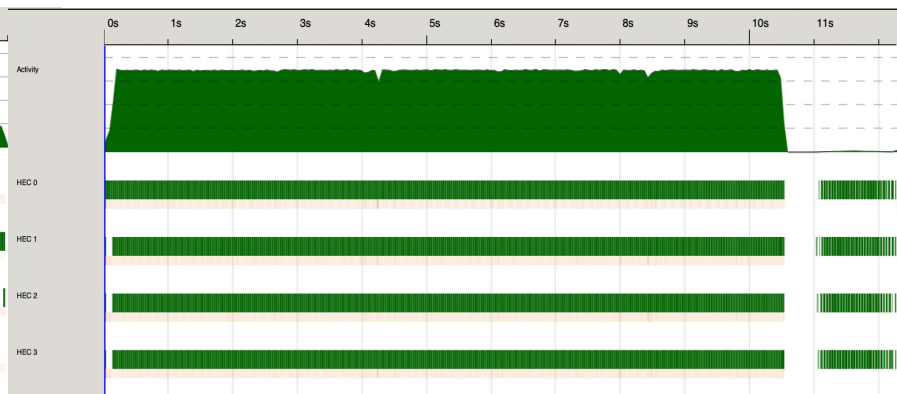
1.8x reduction in GC time

Sequential vs Parallel Implementation

Randomly Generated Data: 300 Nodes over 300 time steps (4 cores)



Total time: 40.649s
Mutator time: 39.108s
GC time: 1.542s
Productivity: 96.2% of mutator vs total



Total time: 12.267s
Mutator time: 11.514s
GC time: 753.437ms
Productivity: 93.9% of mutator vs total

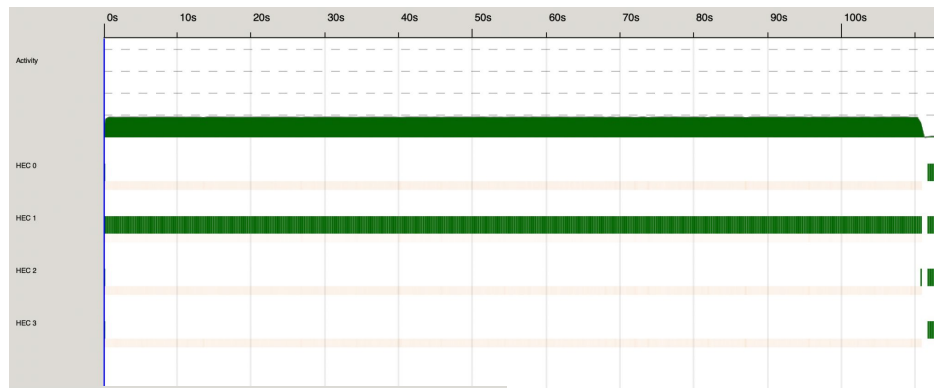
3.31x speedup

3.39x reduction in Mutator Time

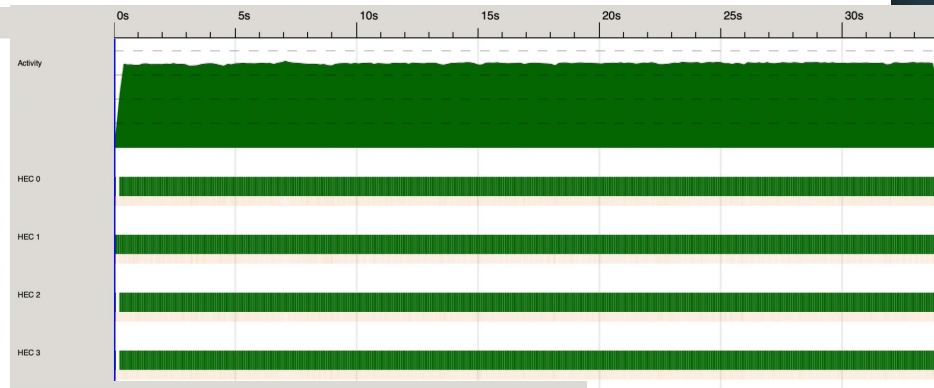
2.05x reduction in GC time

Sequential vs Parallel Implementation

Randomly Generated Data: 500 Nodes over 300 time steps (4 cores)



Total time: 112.873s
Mutator time: 108.758s
GC time: 4.115s
Productivity: 96.4% of mutator vs total

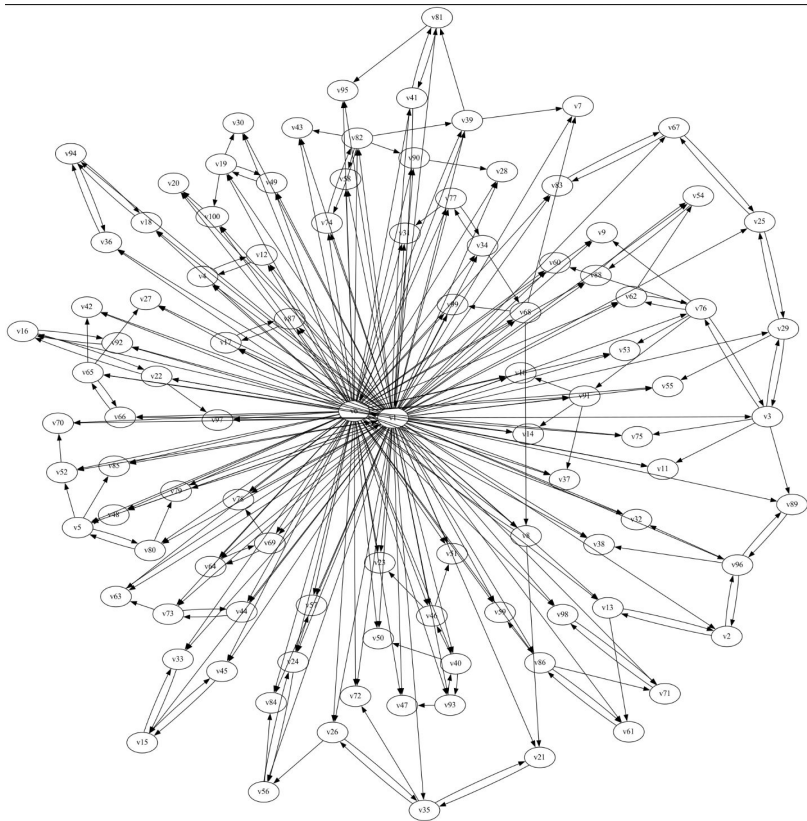


Total time: 35.949s
Mutator time: 33.658s
GC time: 2.292s
Productivity: 93.6% of mutator vs total

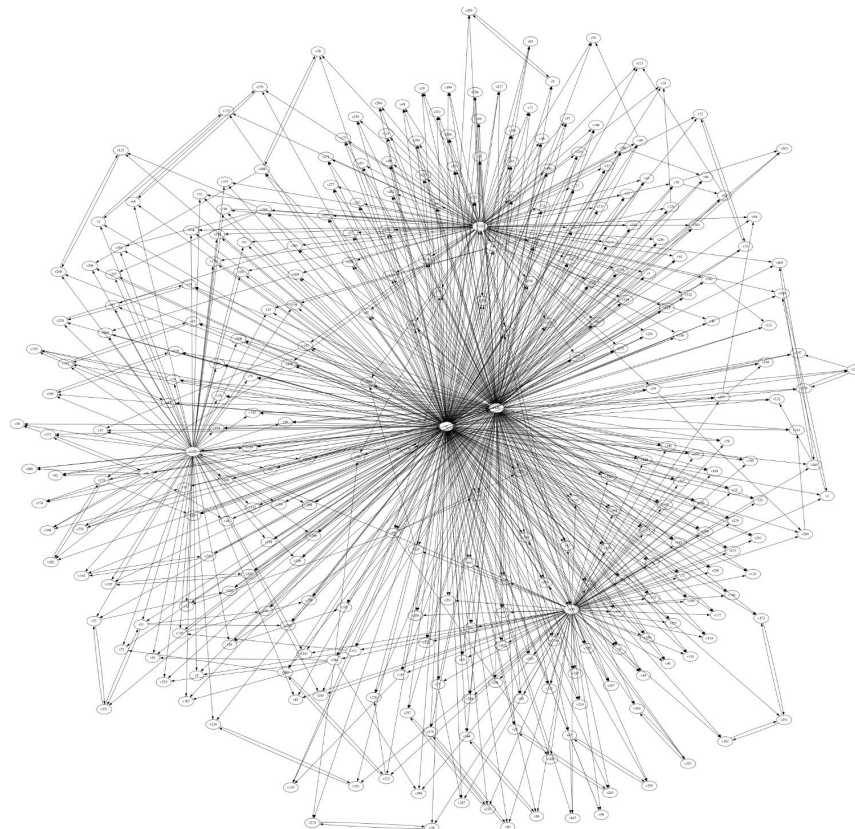
3.23x speedup

3.39x reduction in Mutator Time

2.05x reduction in GC time



100 Nodes over 300 time steps



300 Nodes over 300 time steps