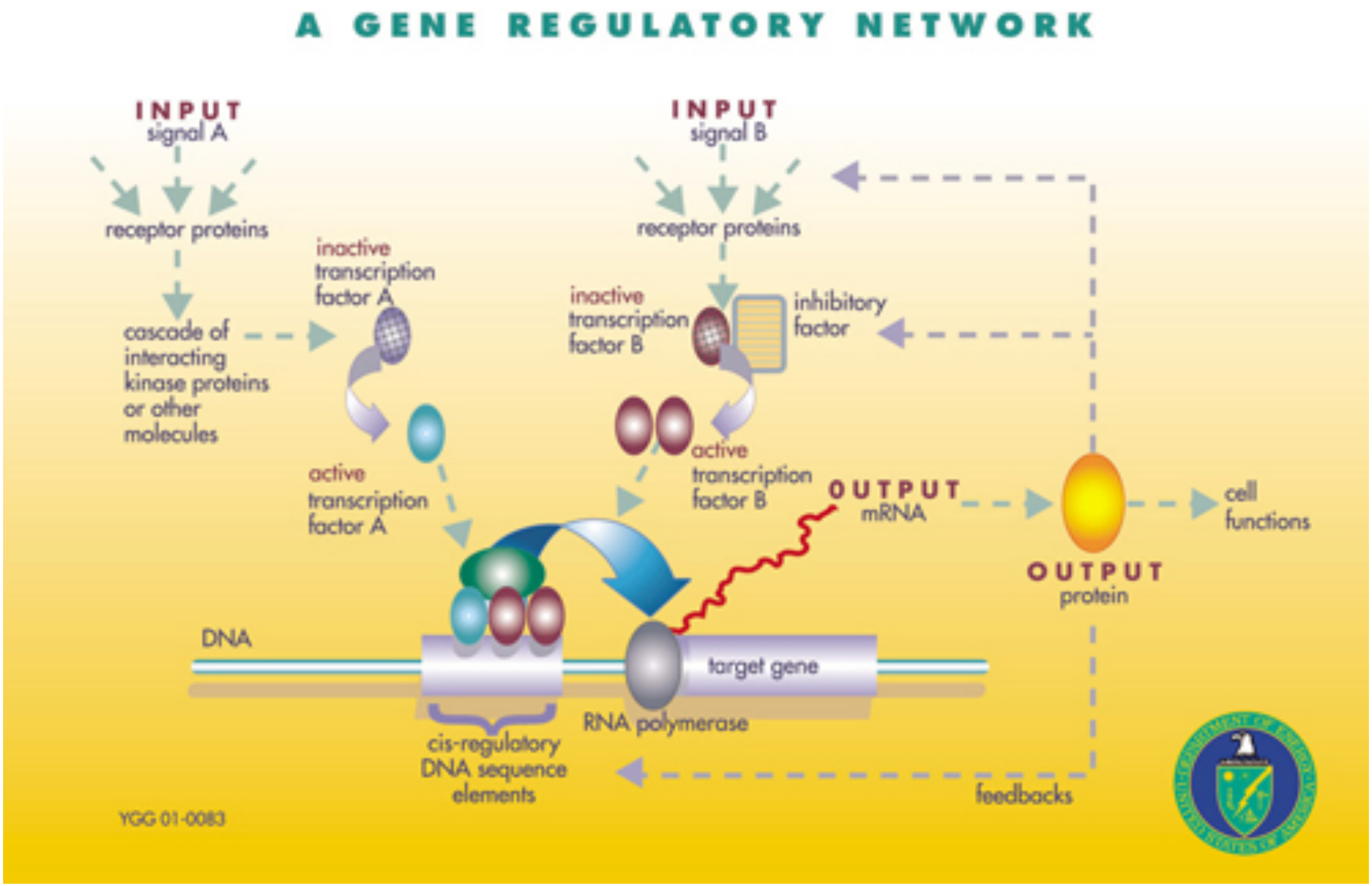
Learning Boolean Models of Regulatory Networks Using Improved Particle Swarm Optimization

Hamim Zafar
Advisor: Prof. Luay Nakhleh

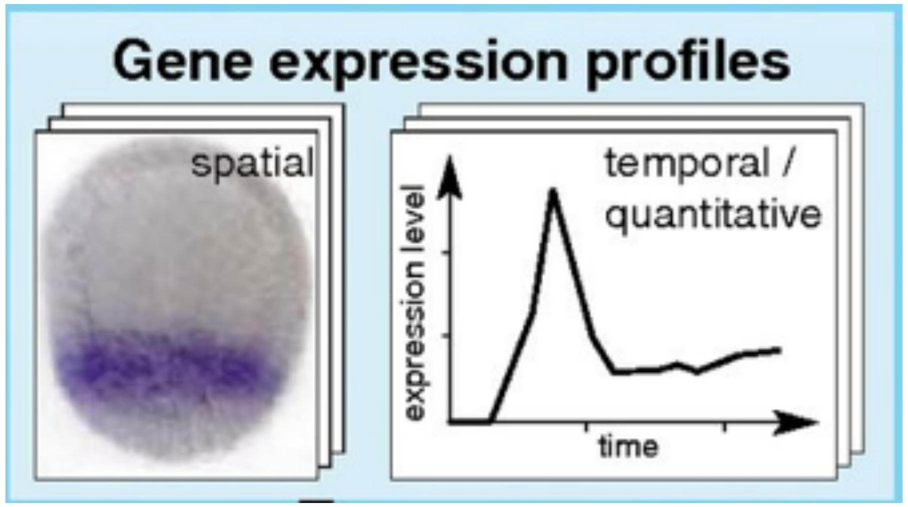
Comp 600
10/28/2013


A GENE REGULATORY NETWORK
INPUT
signal A
receptor proteins
cascade of interacting kinase proteins or other molecules
inactive transcription factor A
active transcription factor A
INPUT
signal B
receptor proteins
inactive transcription factor B
inhibitory factor
active transcription factor B
OUTPUT
mRNA
OUTPUT
protein
cell functions
DNA
target gene
RNA polymerase
cis-regulatory DNA sequence elements
feedbacks
YGG 01-0083

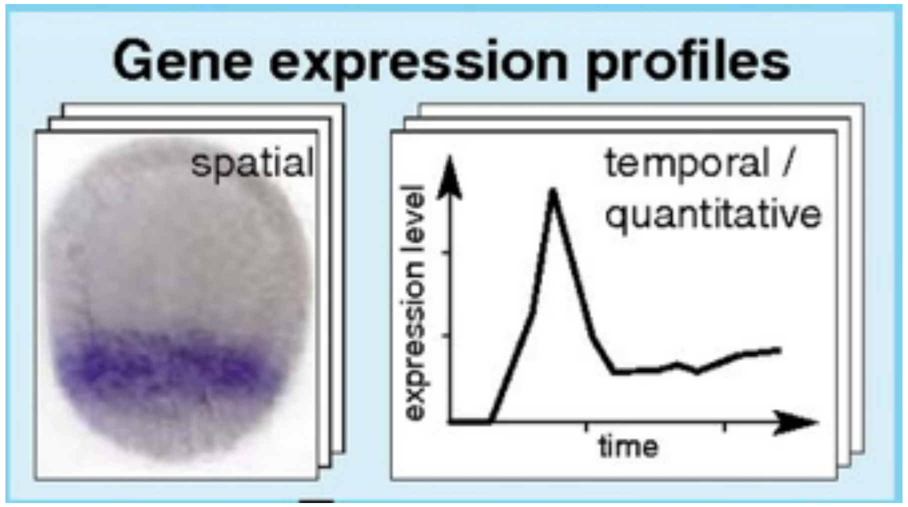
Why are Gene Regulatory Networks (GRN) Important?

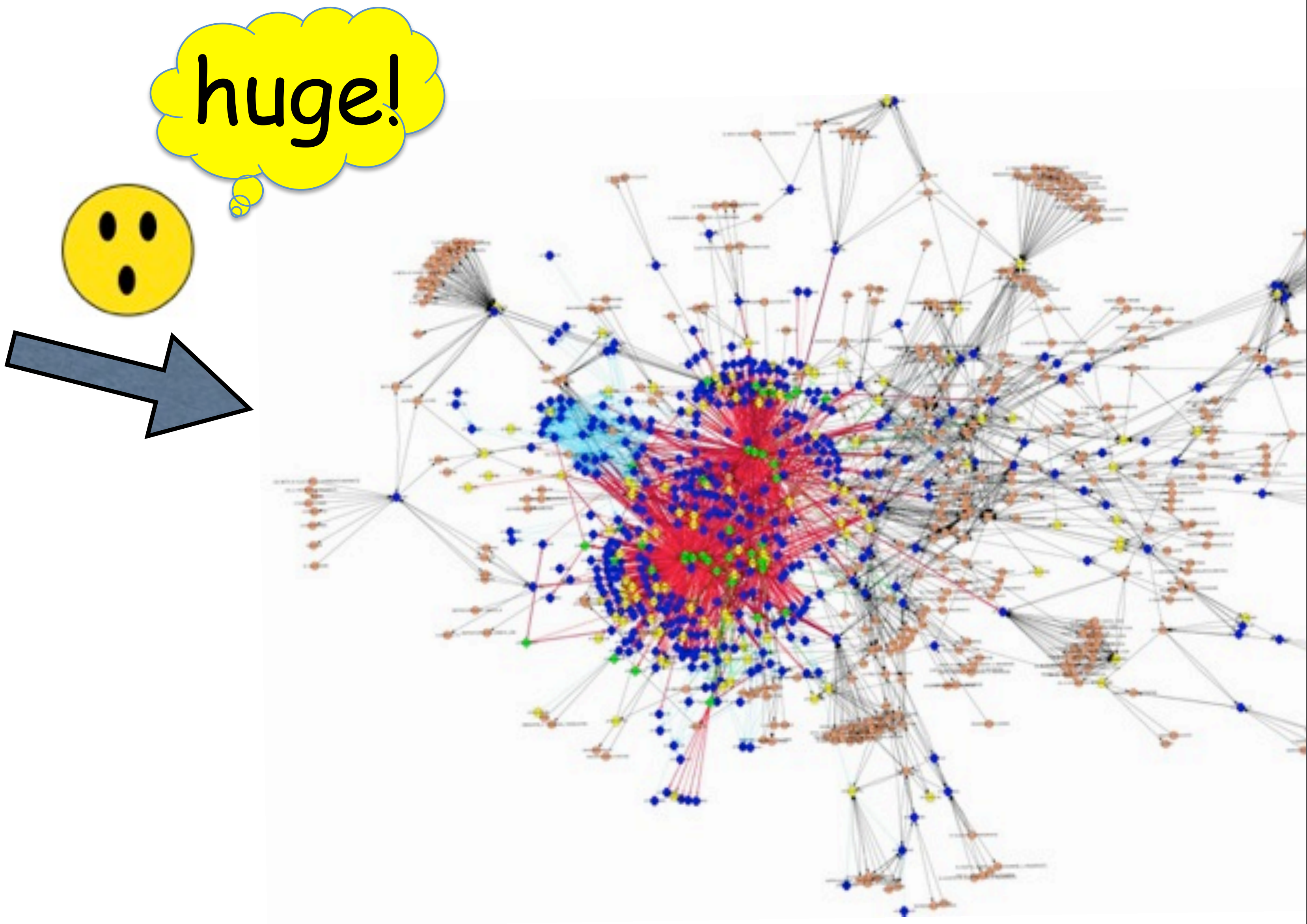
- Essential to the survival and adaptation of the organism.
- Malfunctioning leads to **disease** (e.g: **Cancer, Alzheimer's disease**)
- Potential discovery of triggering mechanism.
- Accurate Diagnosis and proper treatments for disease.
- Targeted **drug development**.

Challenges in Inferring GRN

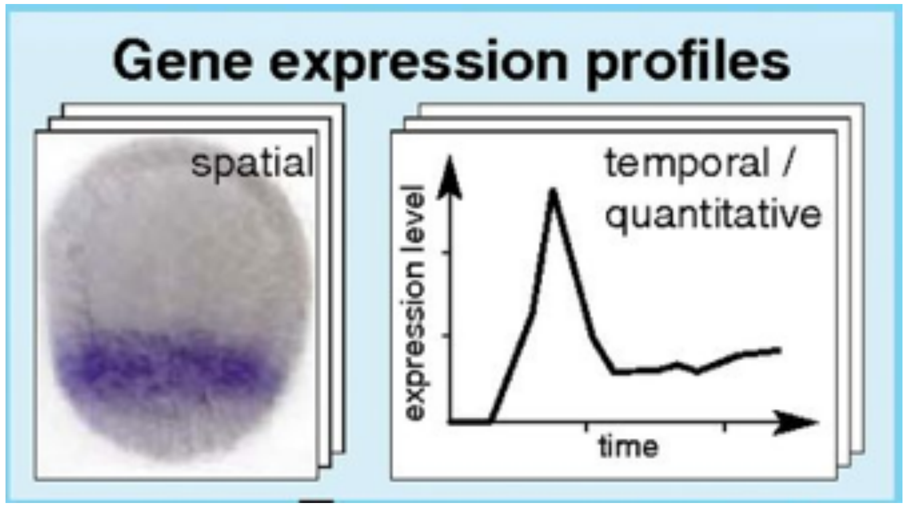


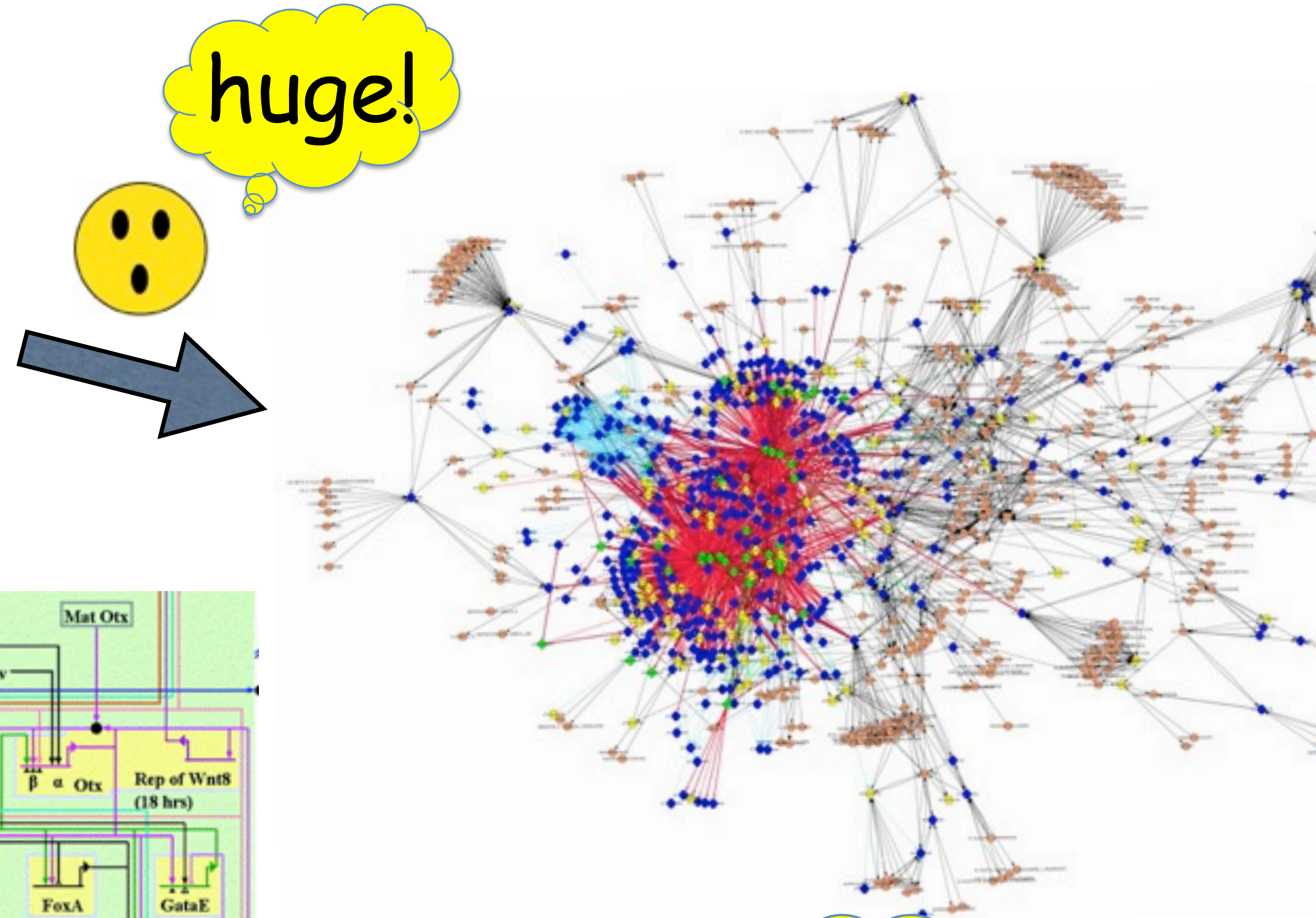
Challenges in Inferring GRN

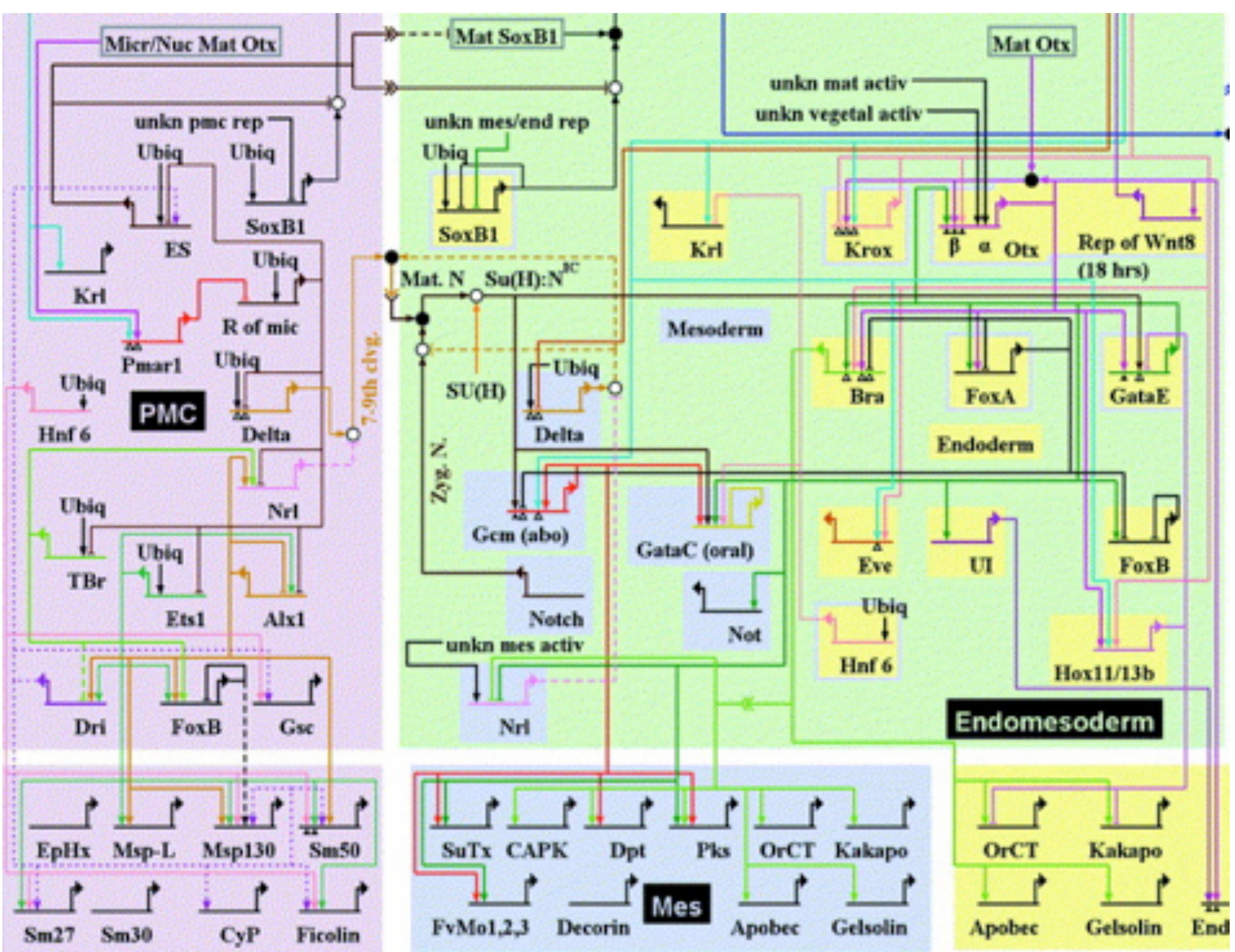




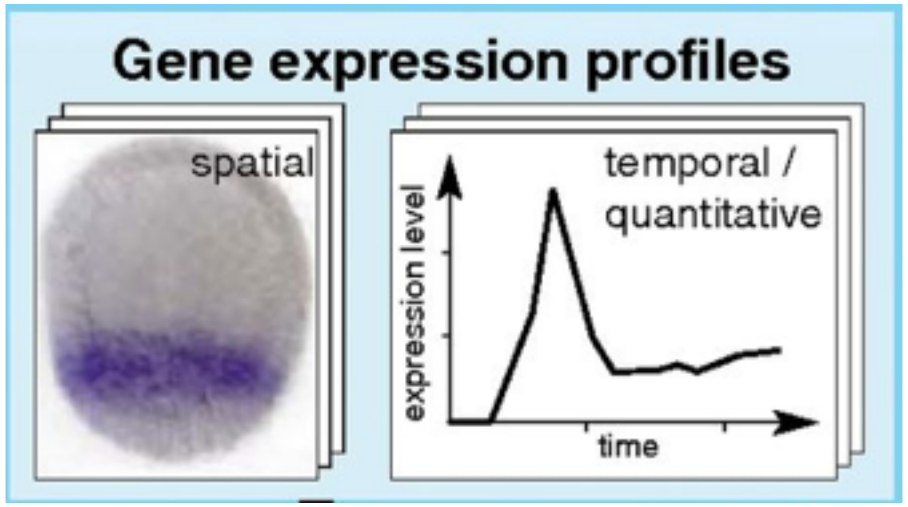
Challenges in Inferring GRN

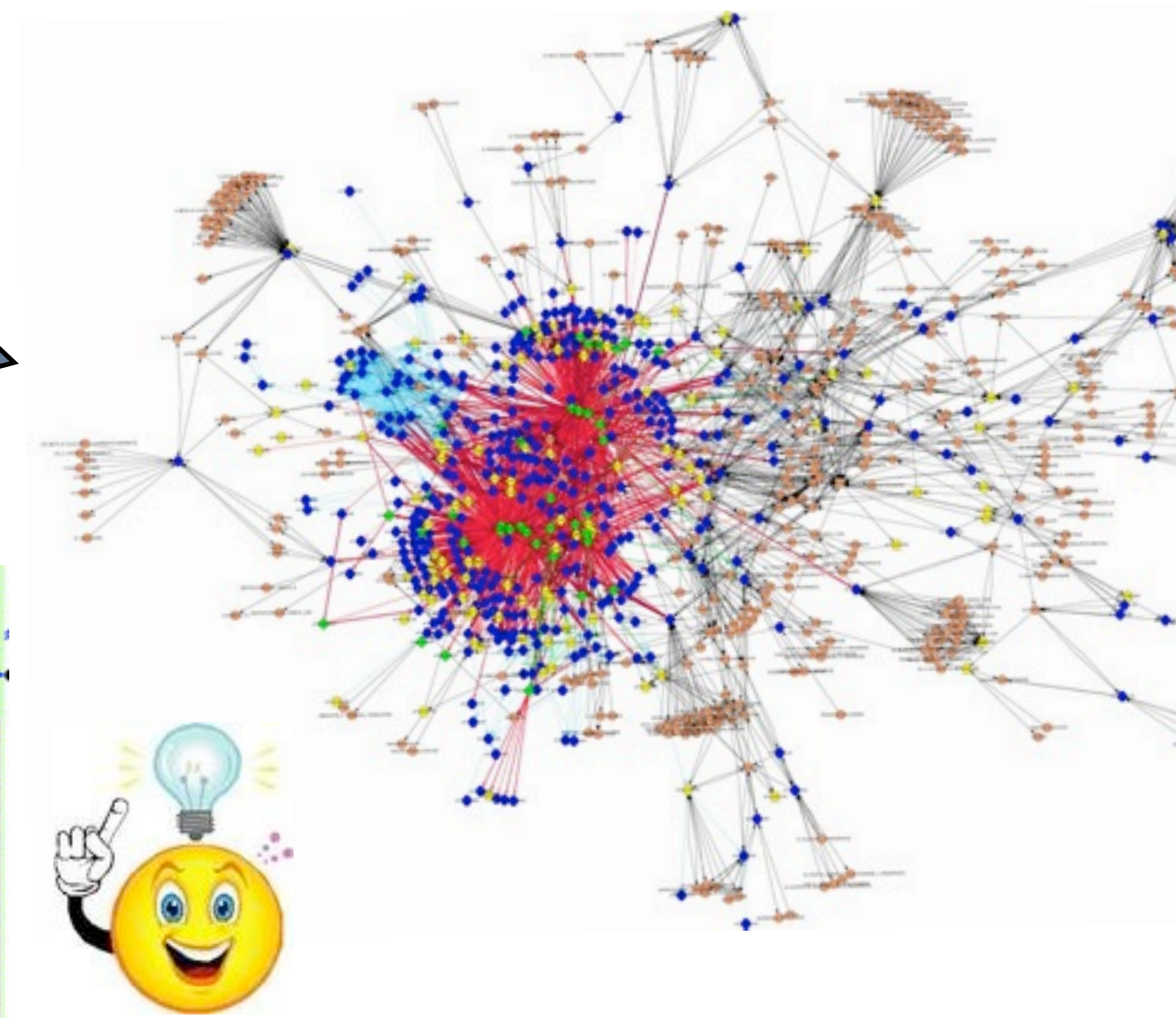


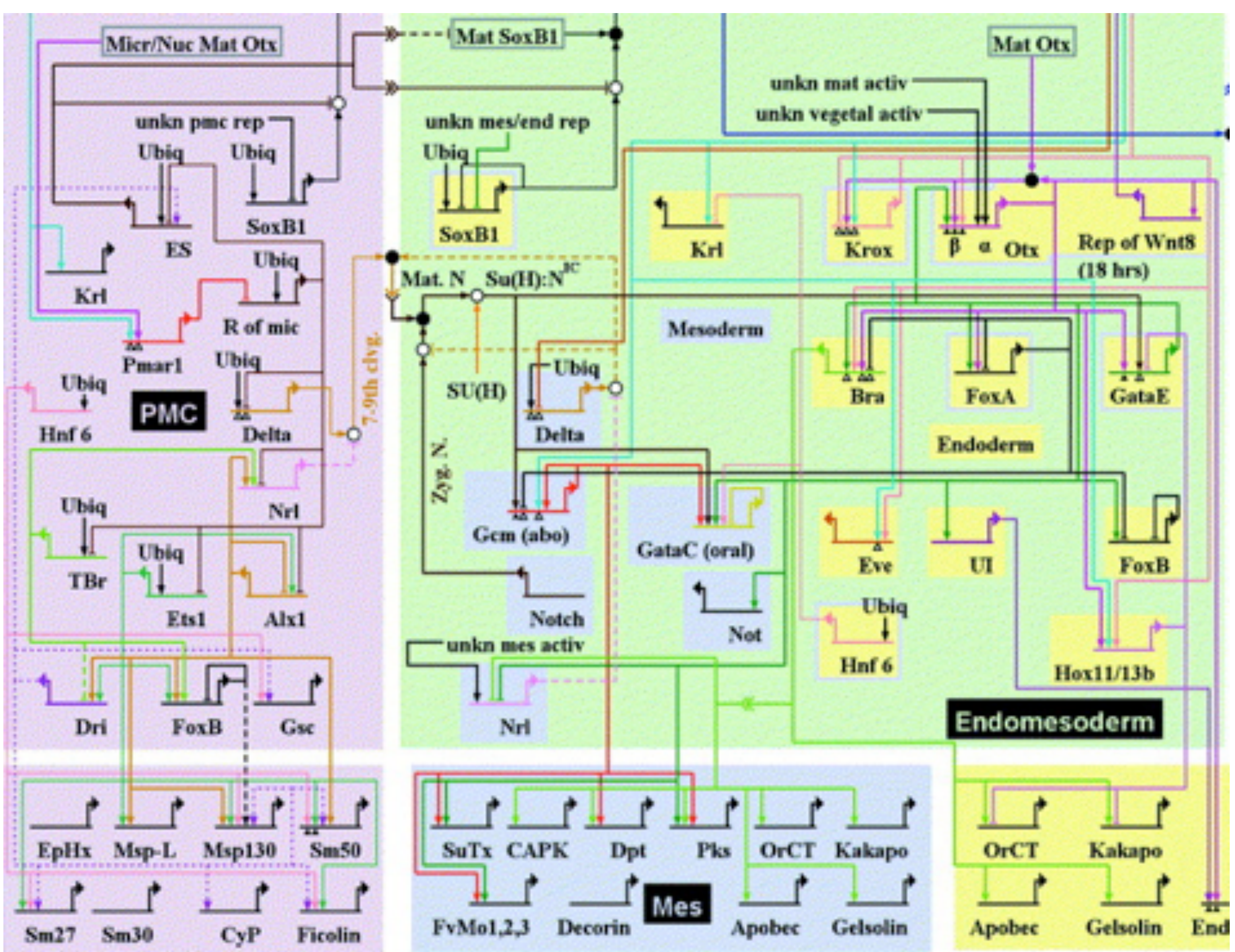




Challenges in Inferring GRN







Remedy:

Modeling Framework

Computational Tool

Modeling Framework for GRN

Linear Models Regression — Lot of Parameters!

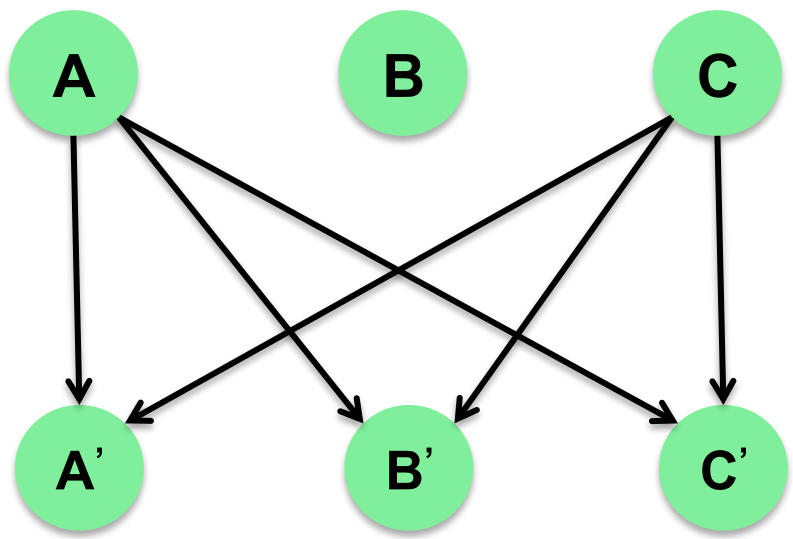
Bayesian Network — No Cycles! No Feedback!

Ordinary Differential Equations — Prior Knowledge of Kinetics??

Boolean Network

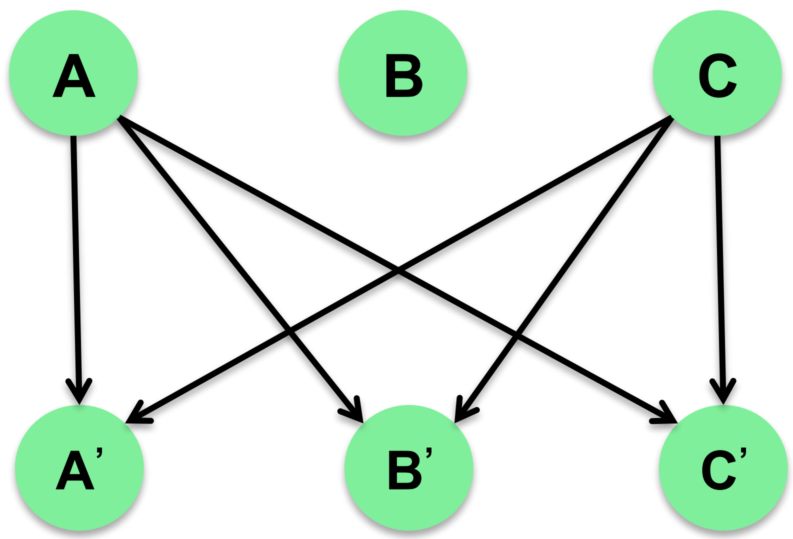
- *Simplicity
- *Capture the Dynamics
- *Capture Steady State Behavior
- *Smaller amount of Data

Boolean Network



A' = not A and C
B' = (not A and not C) or (A and C)
C' = not C or A

Boolean Network

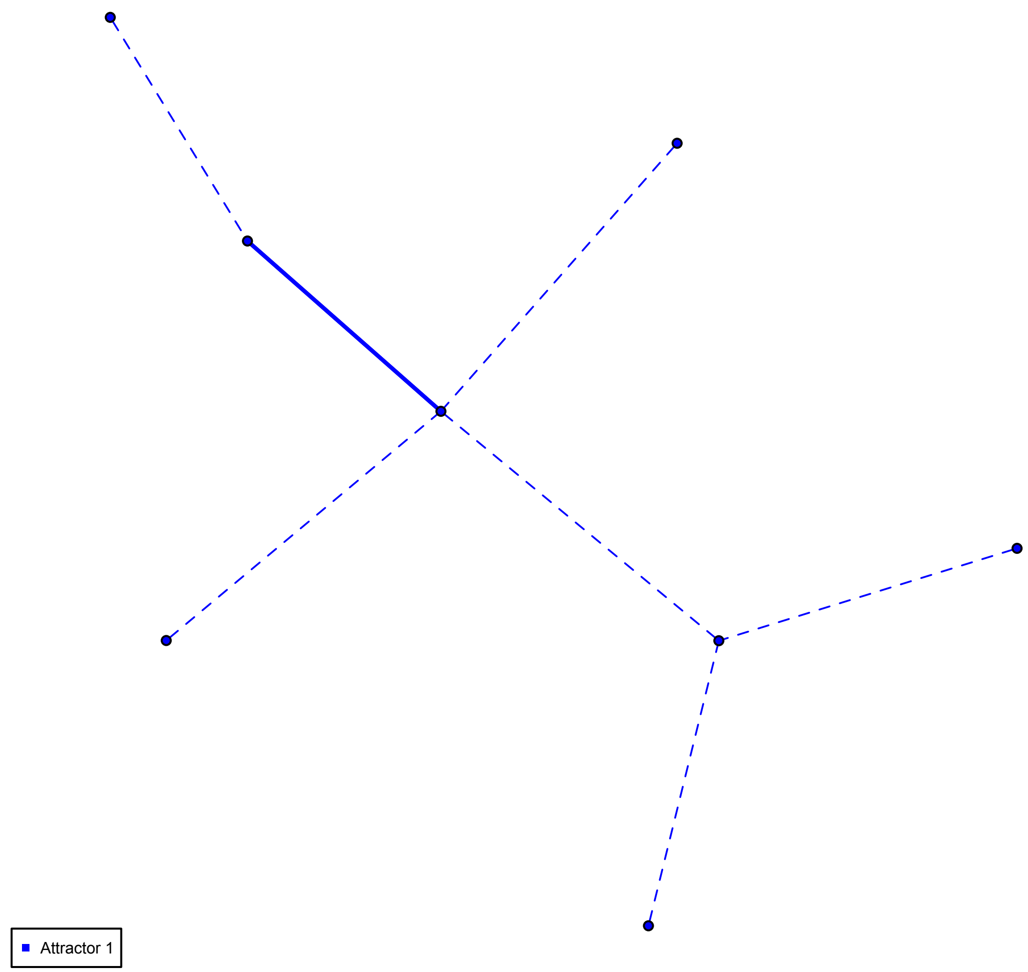


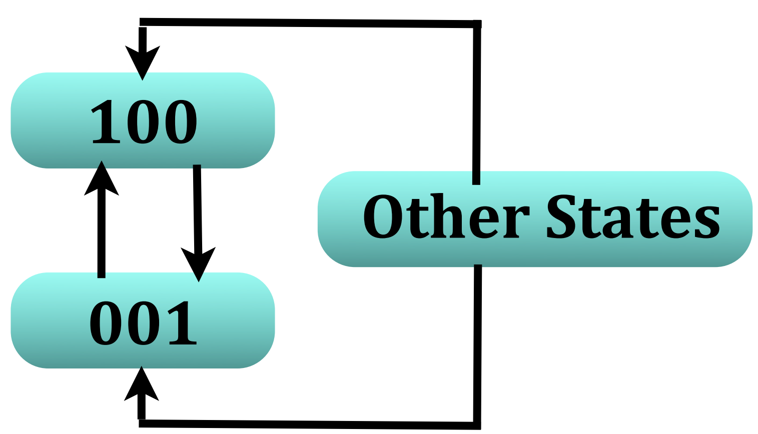
A' = not A and C
B' = (not A and not C) or (A and C)
C' = not C or A

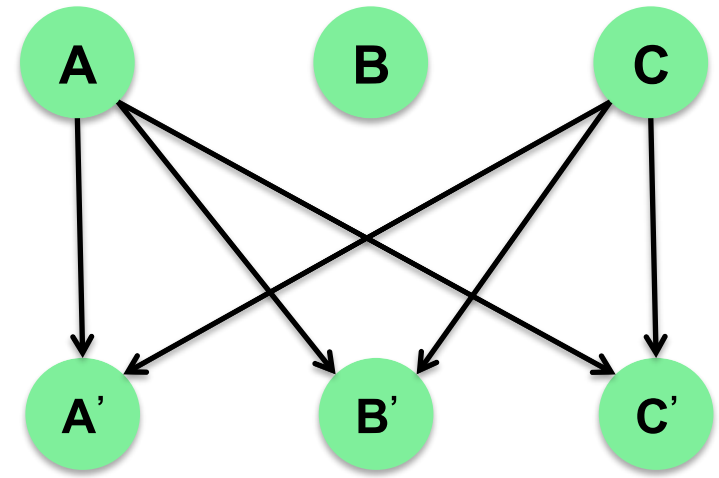
A	B	C	A'	B'	C'
0	0	0	0	1	1
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	0	1	1

- 1 active/expressed
- 0 inactive/unexpressed
- Steady states are called Attractors

Boolean Network



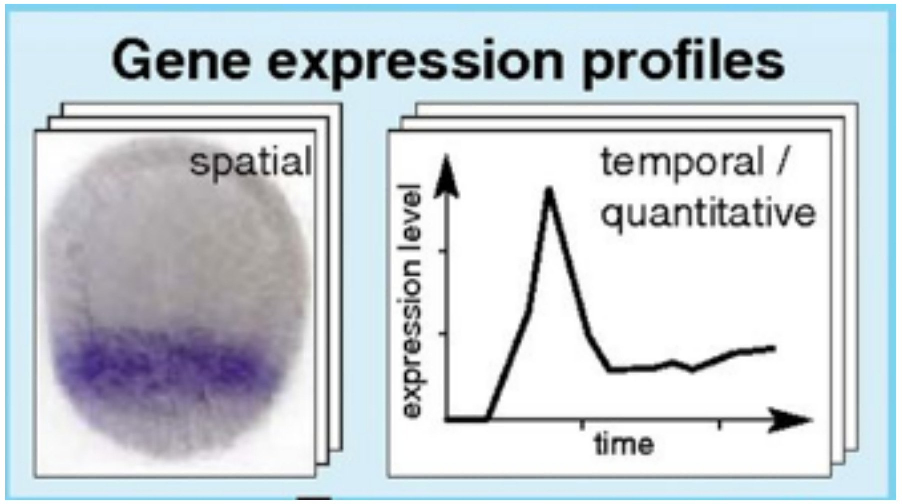


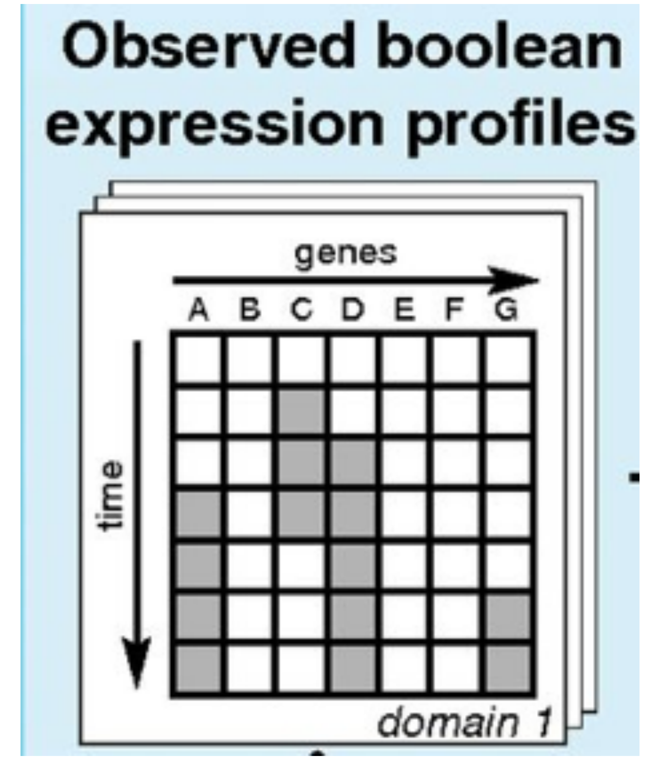


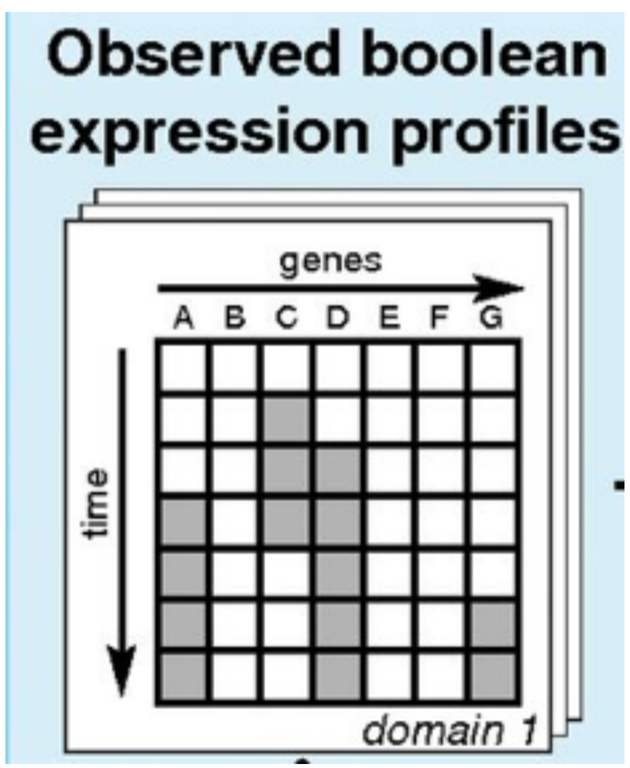
A' = not A and C
B' = (not A and not C) or (A and C)
C' = not C or A

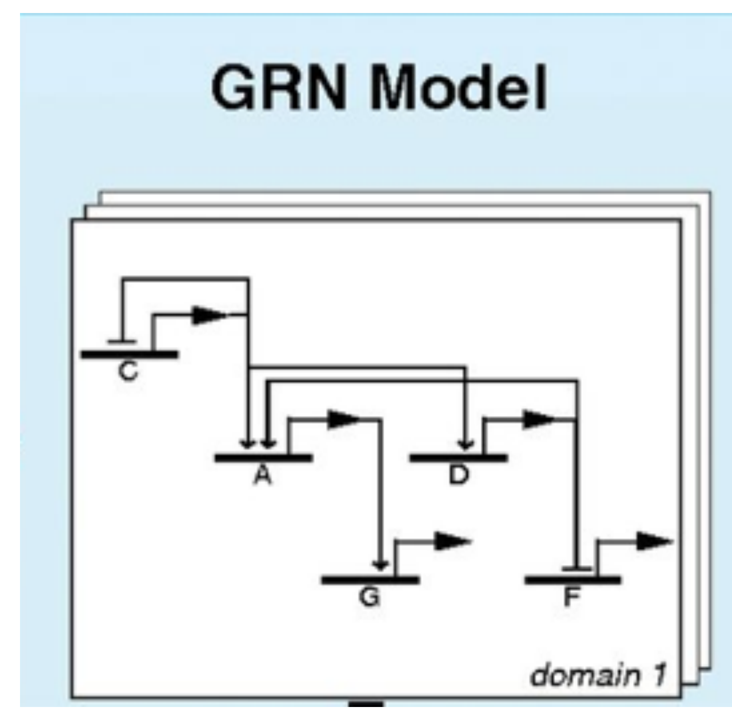
A	B	C	A'	B'	C'
0	0	0	0	1	1
0	0	1	1	0	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	0	1	1

Inference of GRN

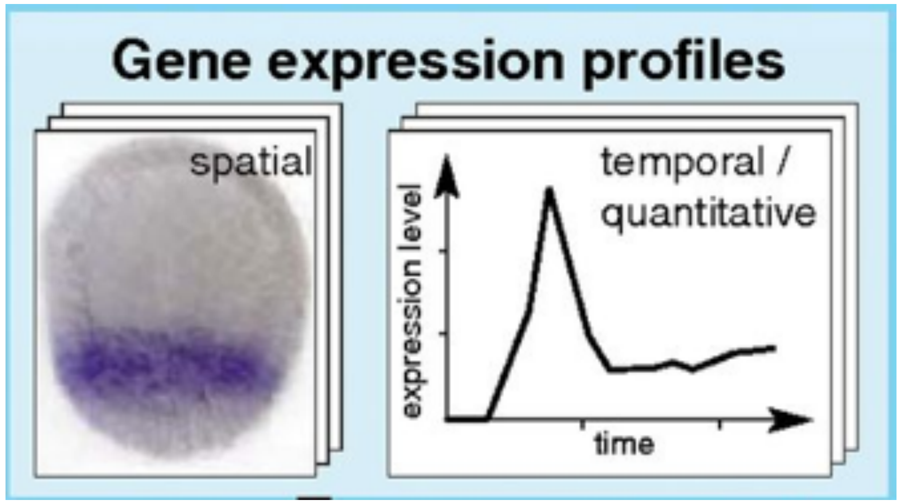






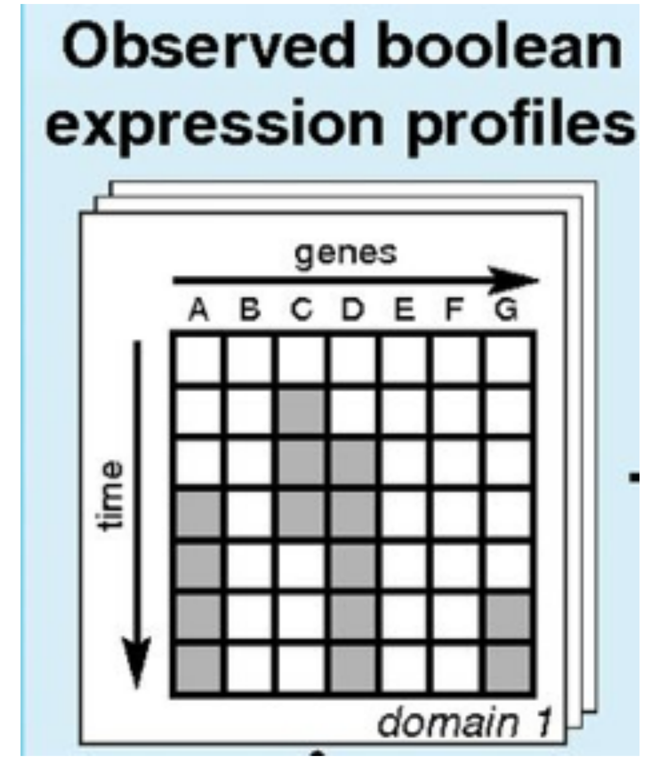


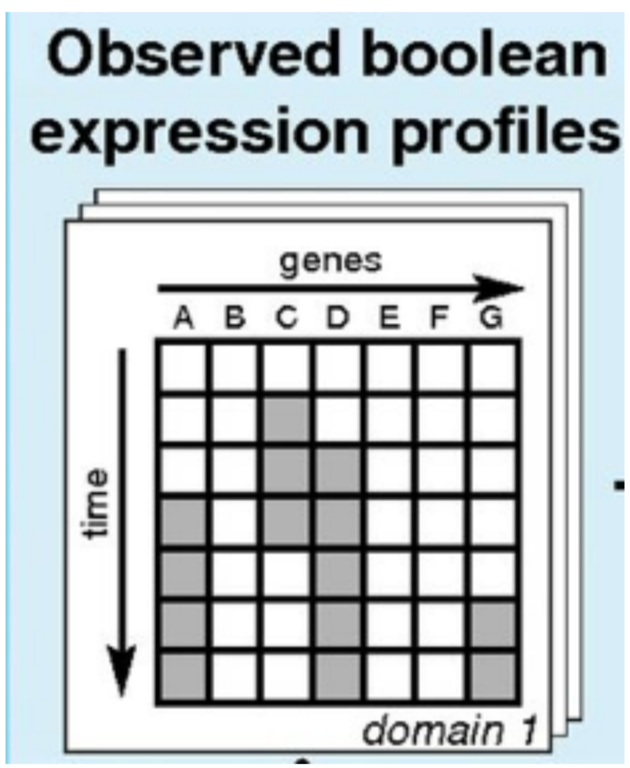
Inference of GRN

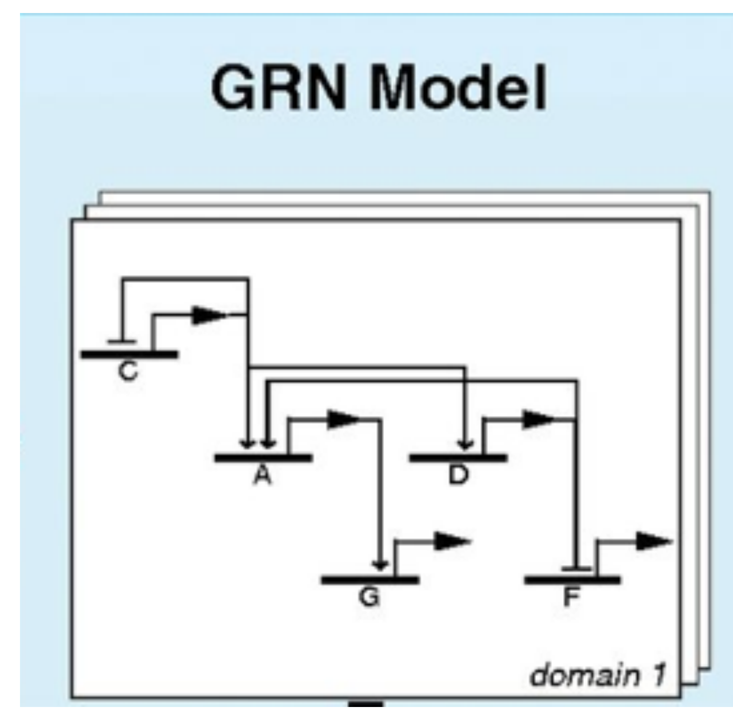


Step 1: Binarization

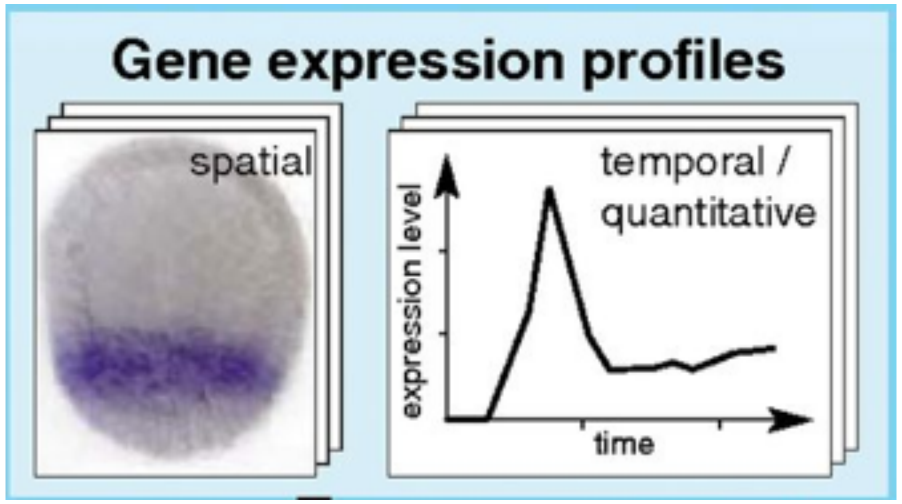
Iterative Clustering





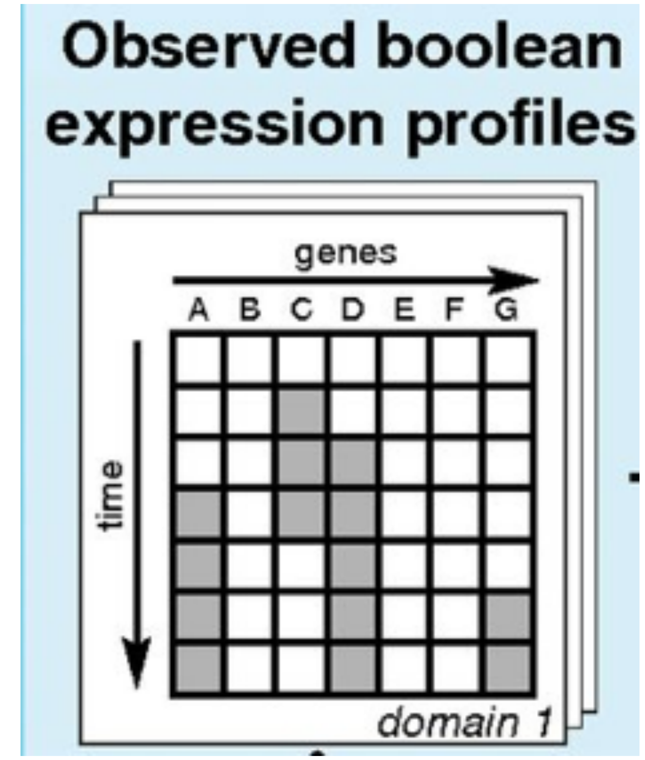


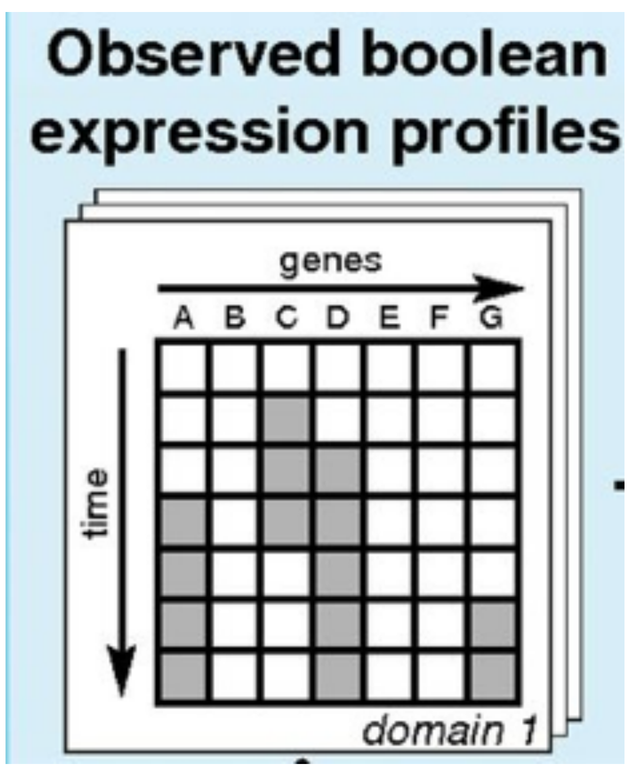
Inference of GRN



Step 1: Binarization

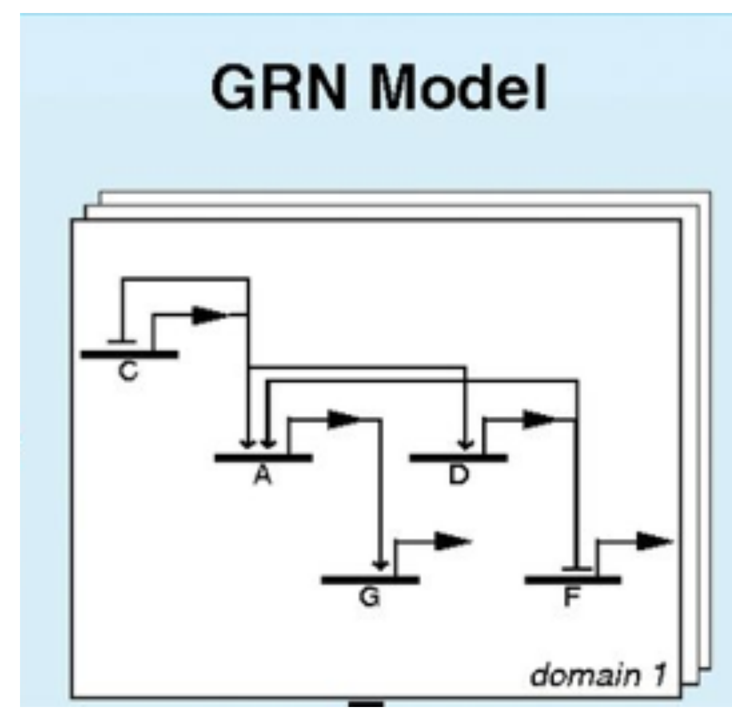
Iterative Clustering





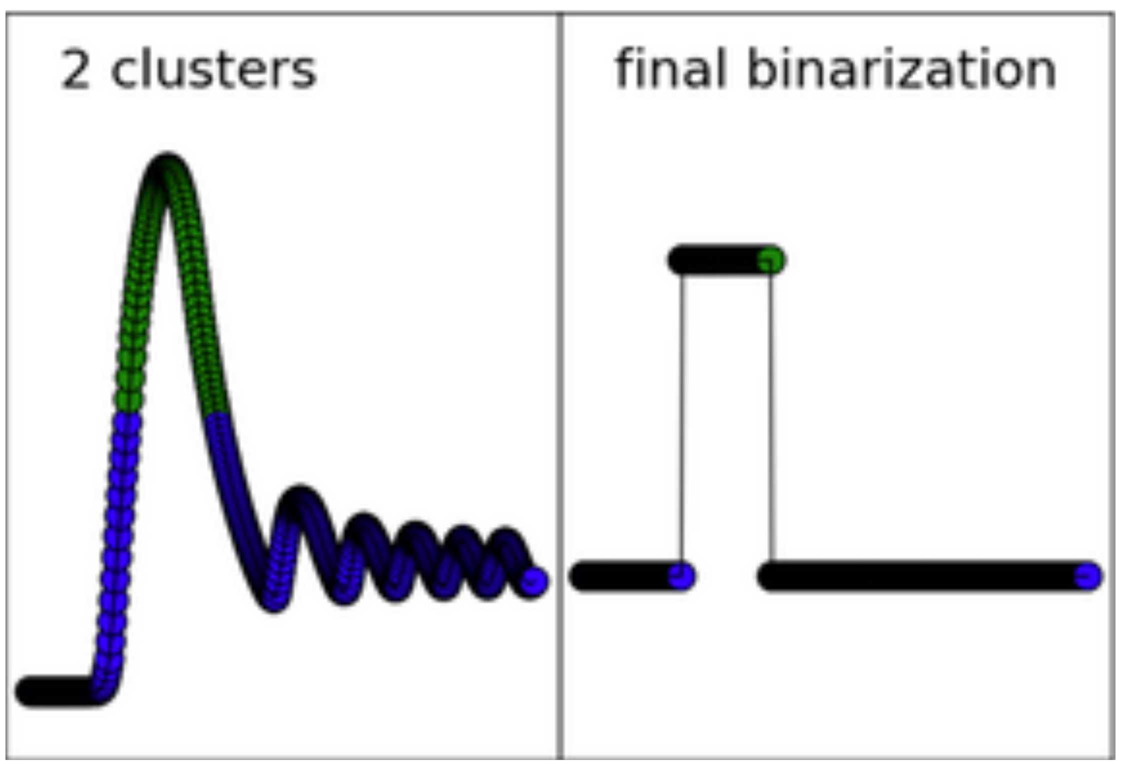
Step 2: Learning

Binary Dynamic Lbest PSO



Binarization using Iterative Clustering

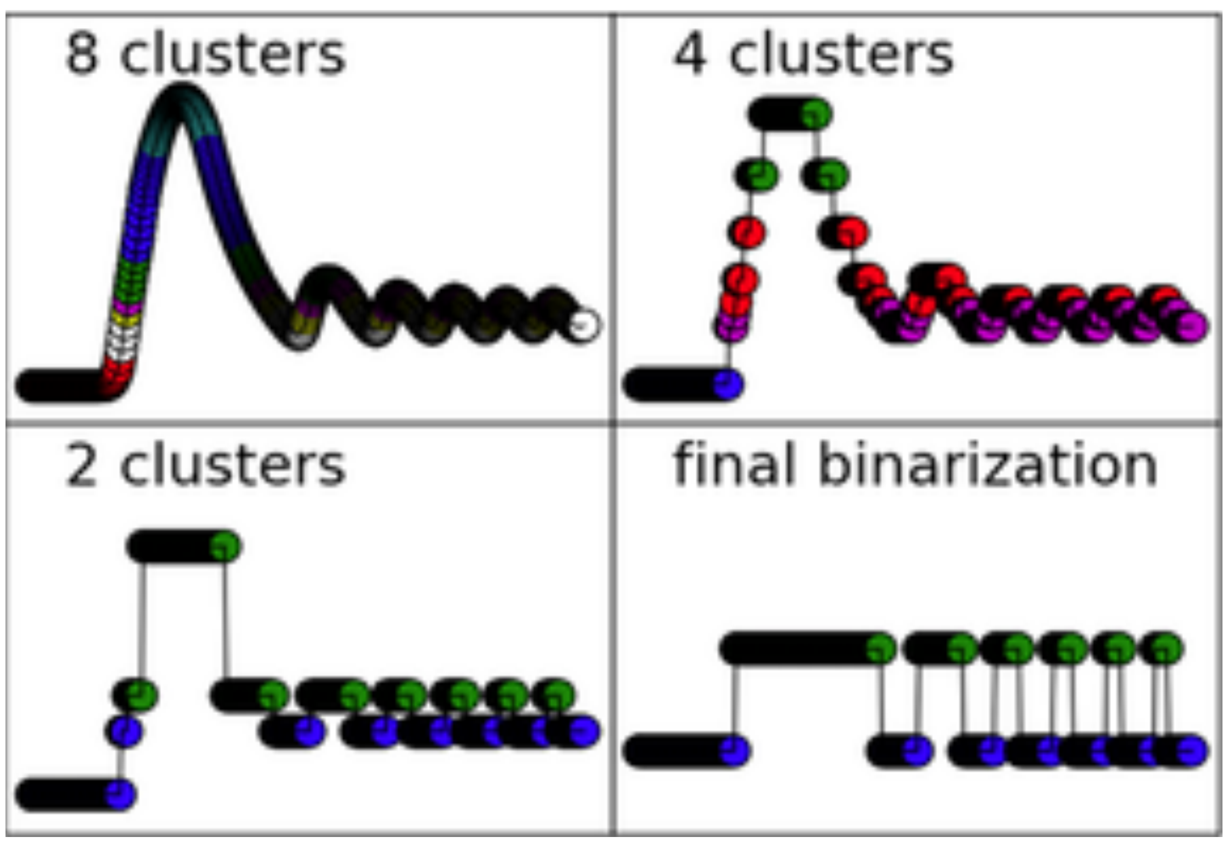
- Input : Time series data for n genes
- Output : Binary time series data
- We want to retain all the information from original data (eg. **Oscillatory dynamics**)



Direct Binarization

Binarization using Iterative Clustering

- Input : Time series data for n genes
- Output : Binary time series data
- We want to retain all the information from original data (eg. **Oscillatory dynamics**)



Iterative Binarization

δ **Depth of Clustering**

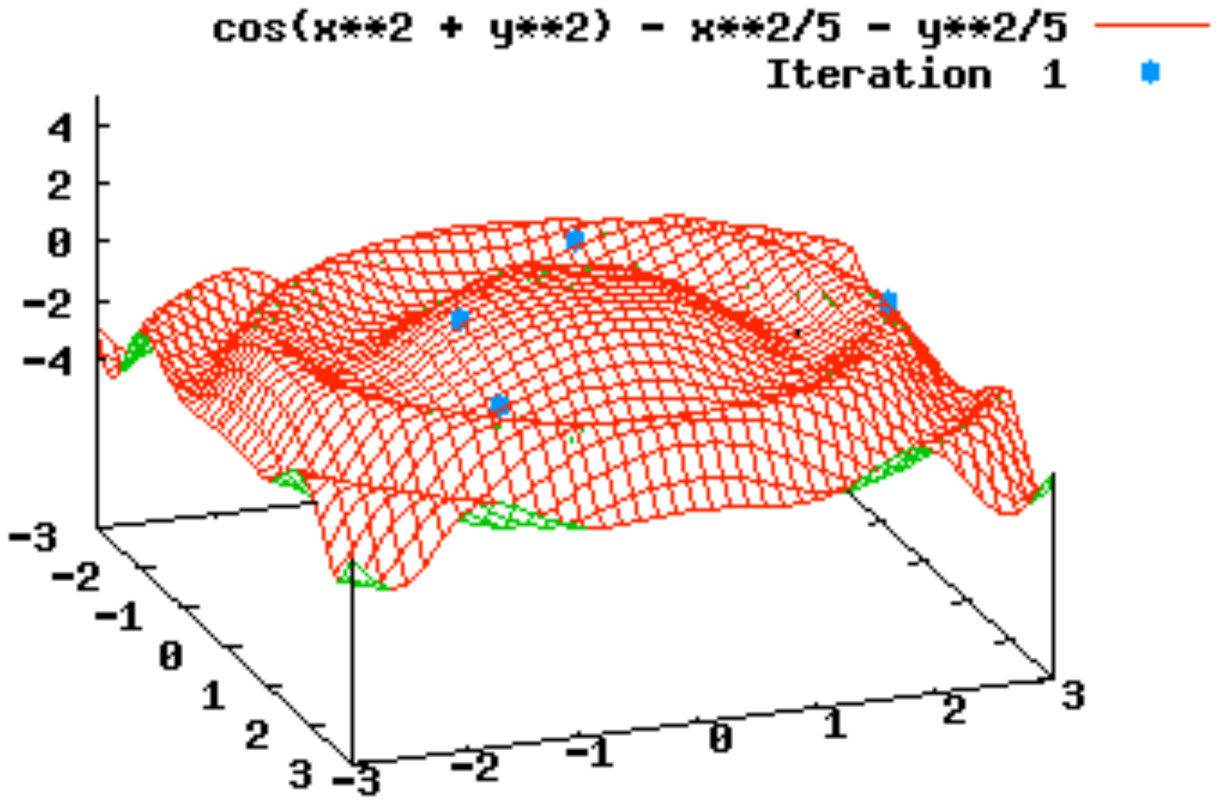
Reference: Berestovsky and Nakhleh 2013

Particle Swarm Optimization (PSO)

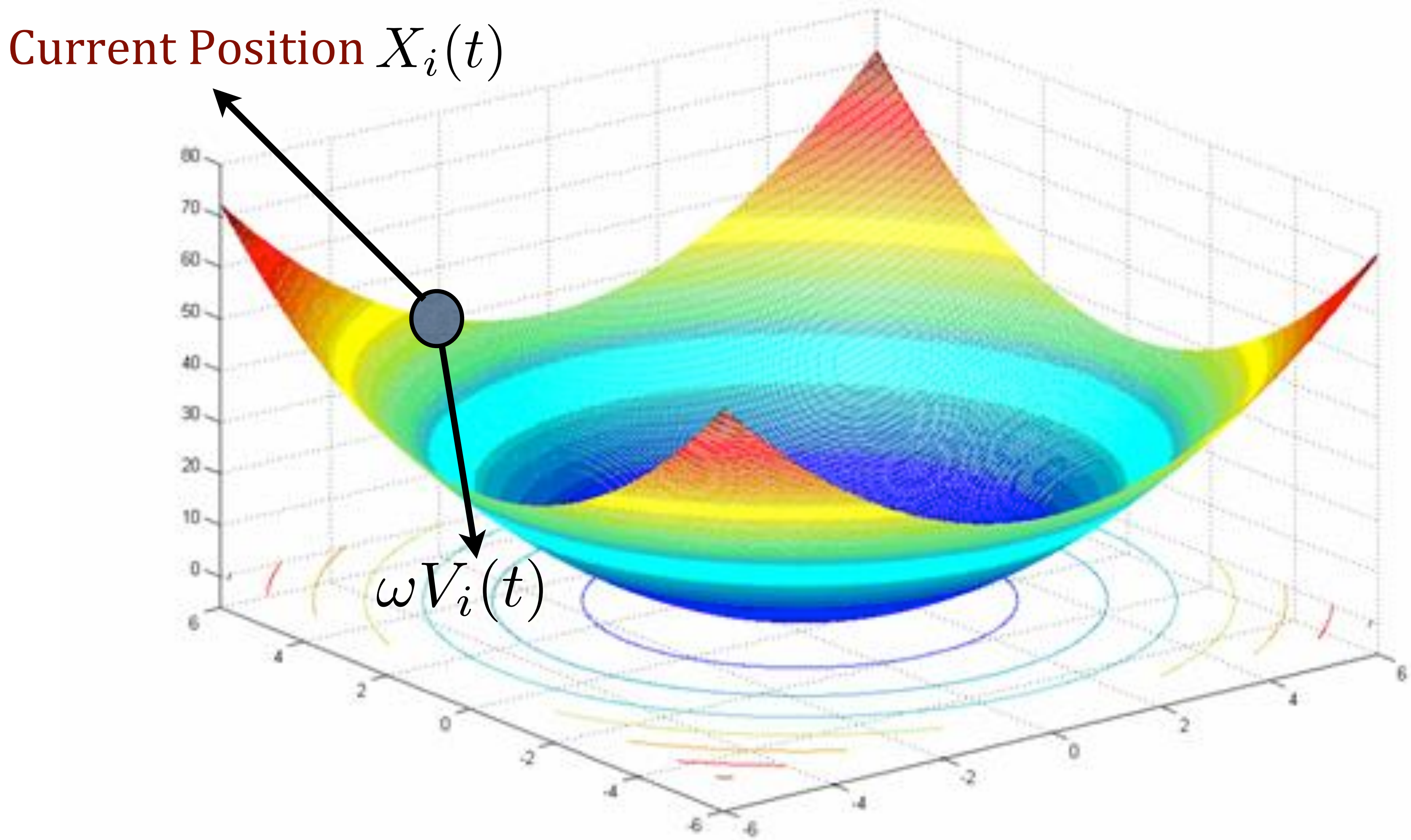
- Nature-inspired Stochastic Search algorithm.
- Kennedy and Eberhart
- Swarm of particles move around search space looking for best solution.
- Combination of **cognitive** and **social learning.**

Particle Swarm Optimization (PSO)

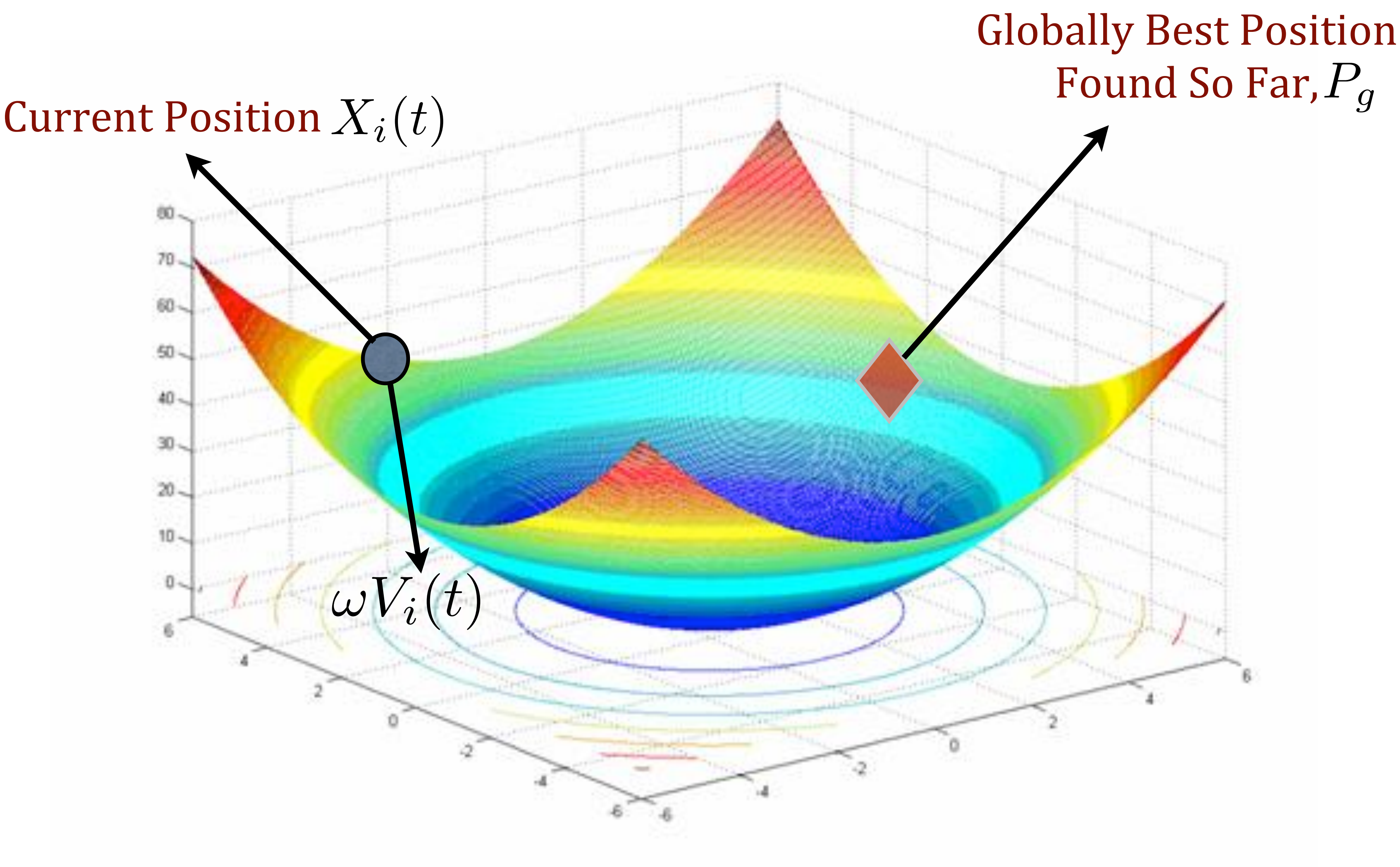
- Nature-inspired Stochastic Search algorithm.
- Kennedy and Eberhart
- Swarm of particles move around search space looking for best solution.
- Combination of **cognitive** and **social learning.**



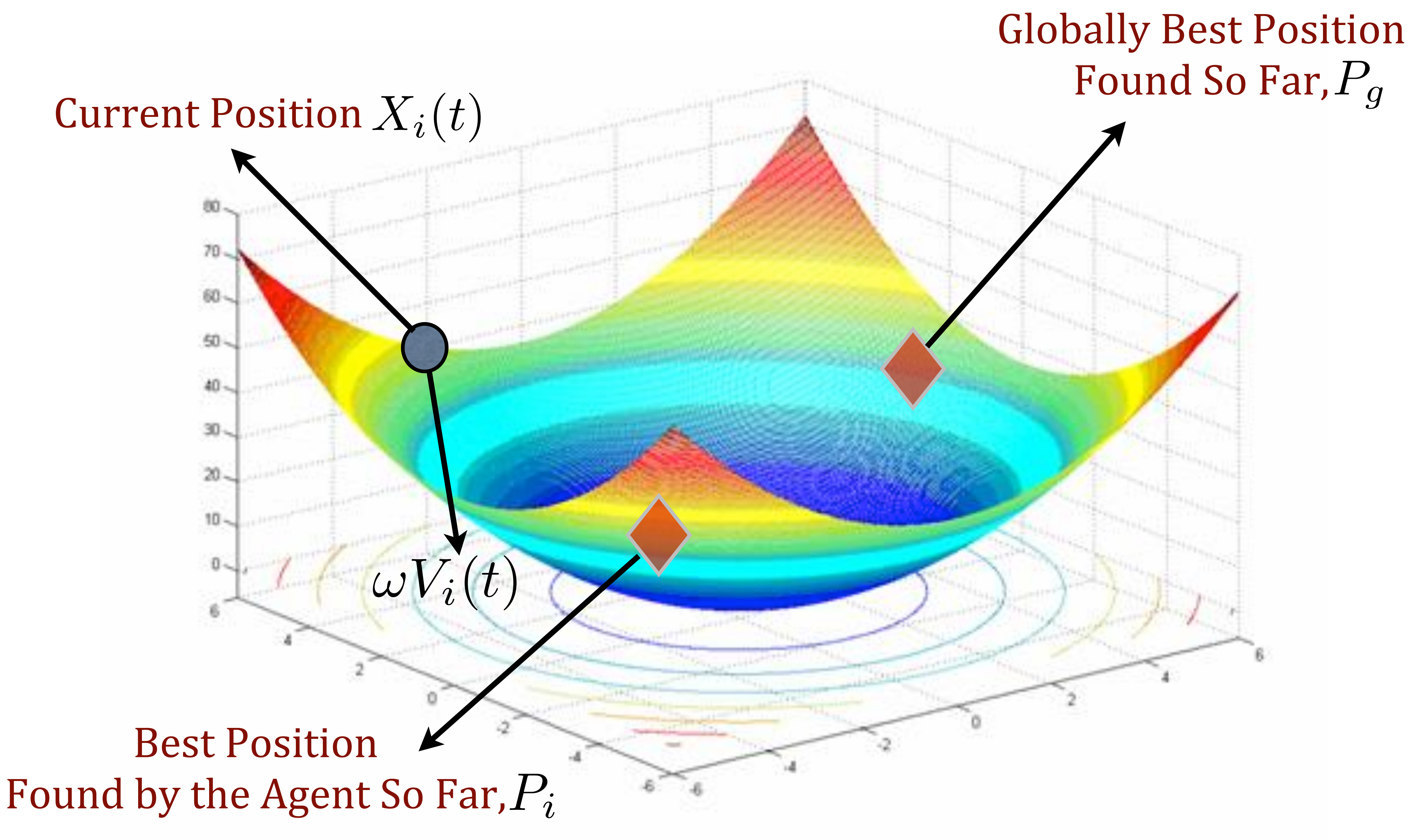
How Cognitive and Social Learning are Combined?



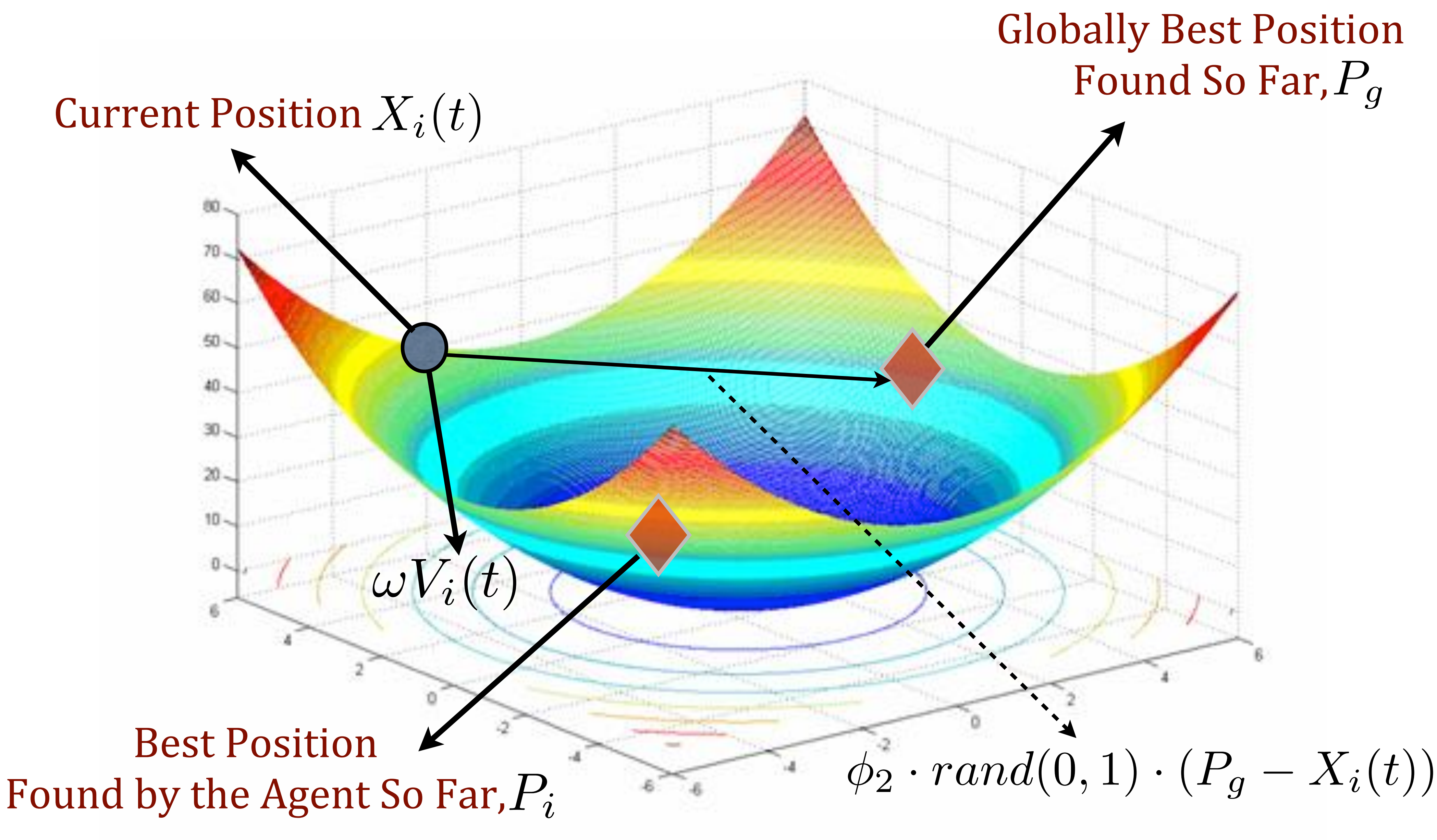
How Cognitive and Social Learning are Combined?



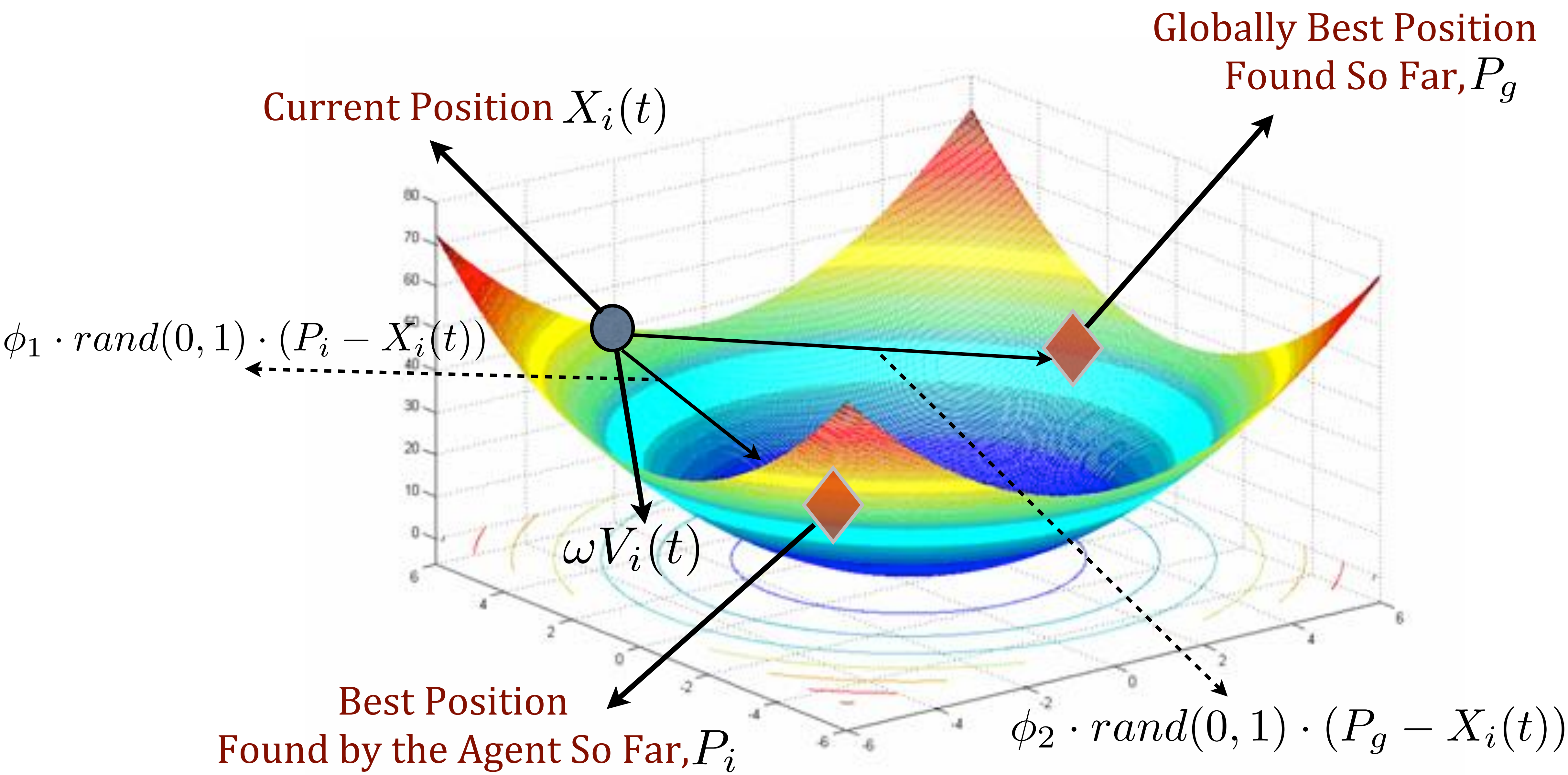
How Cognitive and Social Learning are Combined?



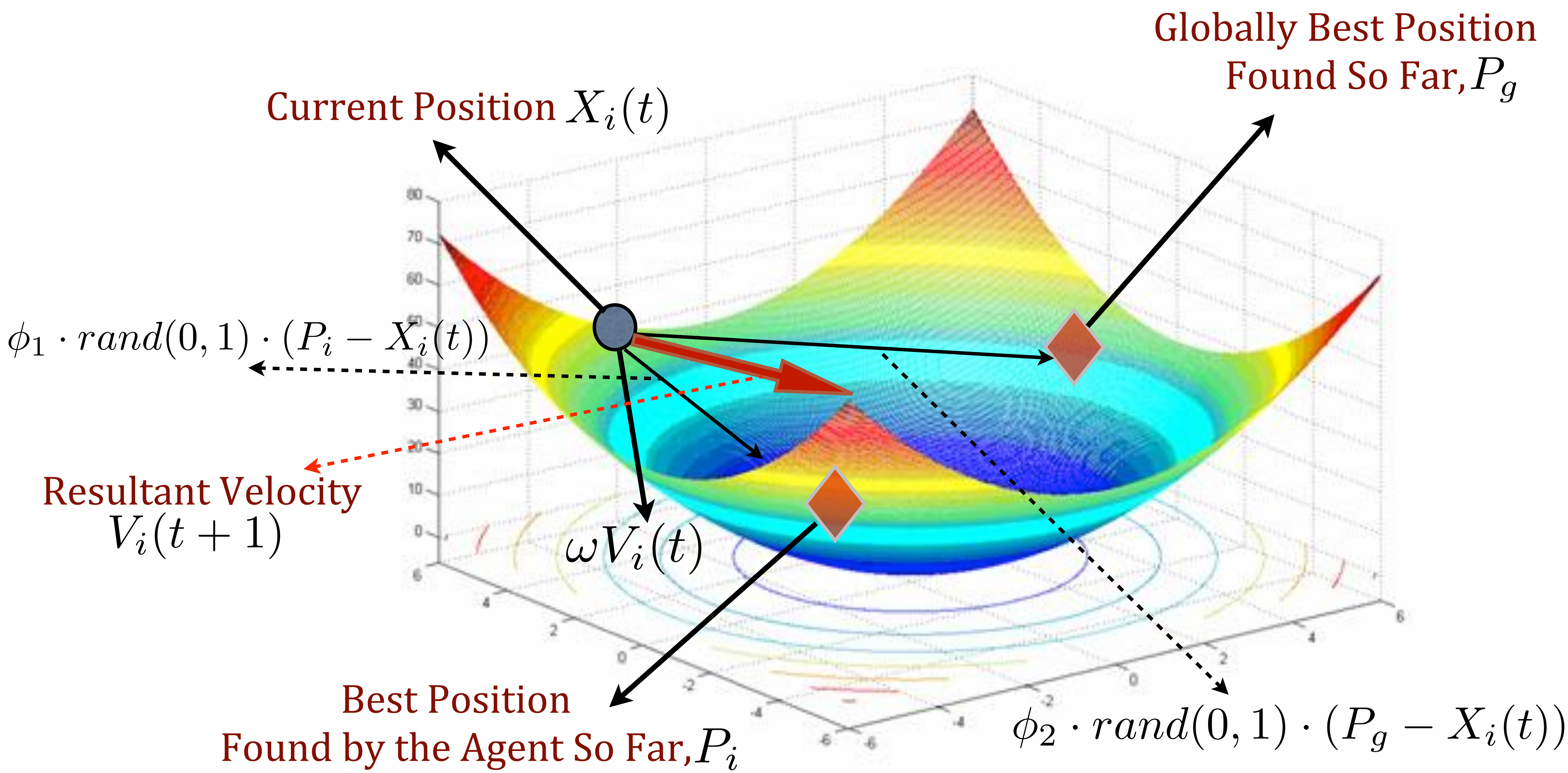
How Cognitive and Social Learning are Combined?



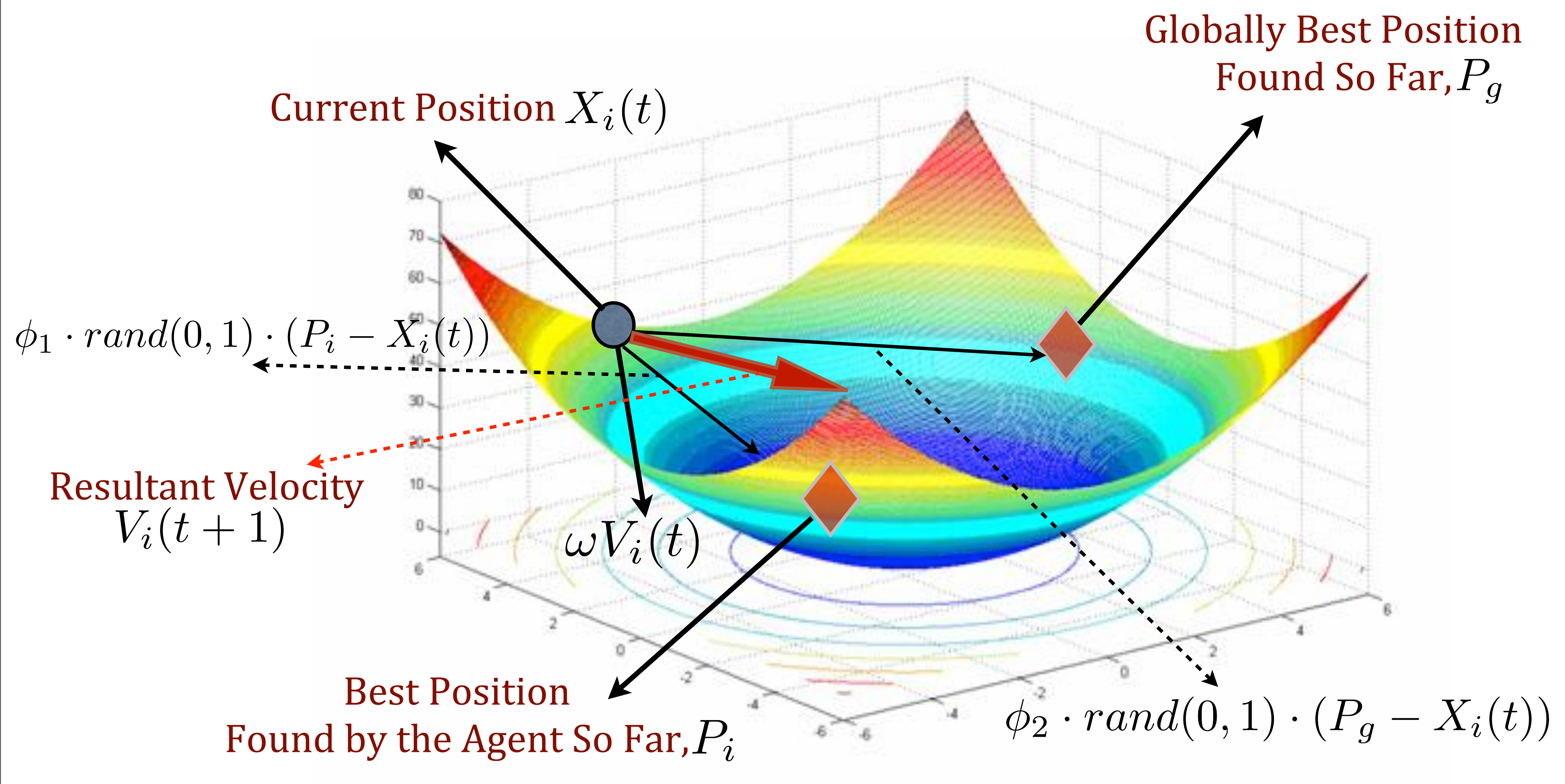
How Cognitive and Social Learning are Combined?



How Cognitive and Social Learning are Combined?



How Cognitive and Social Learning are Combined?



$$X_i(t+1) = X_i(t) + V_i(t+1)$$

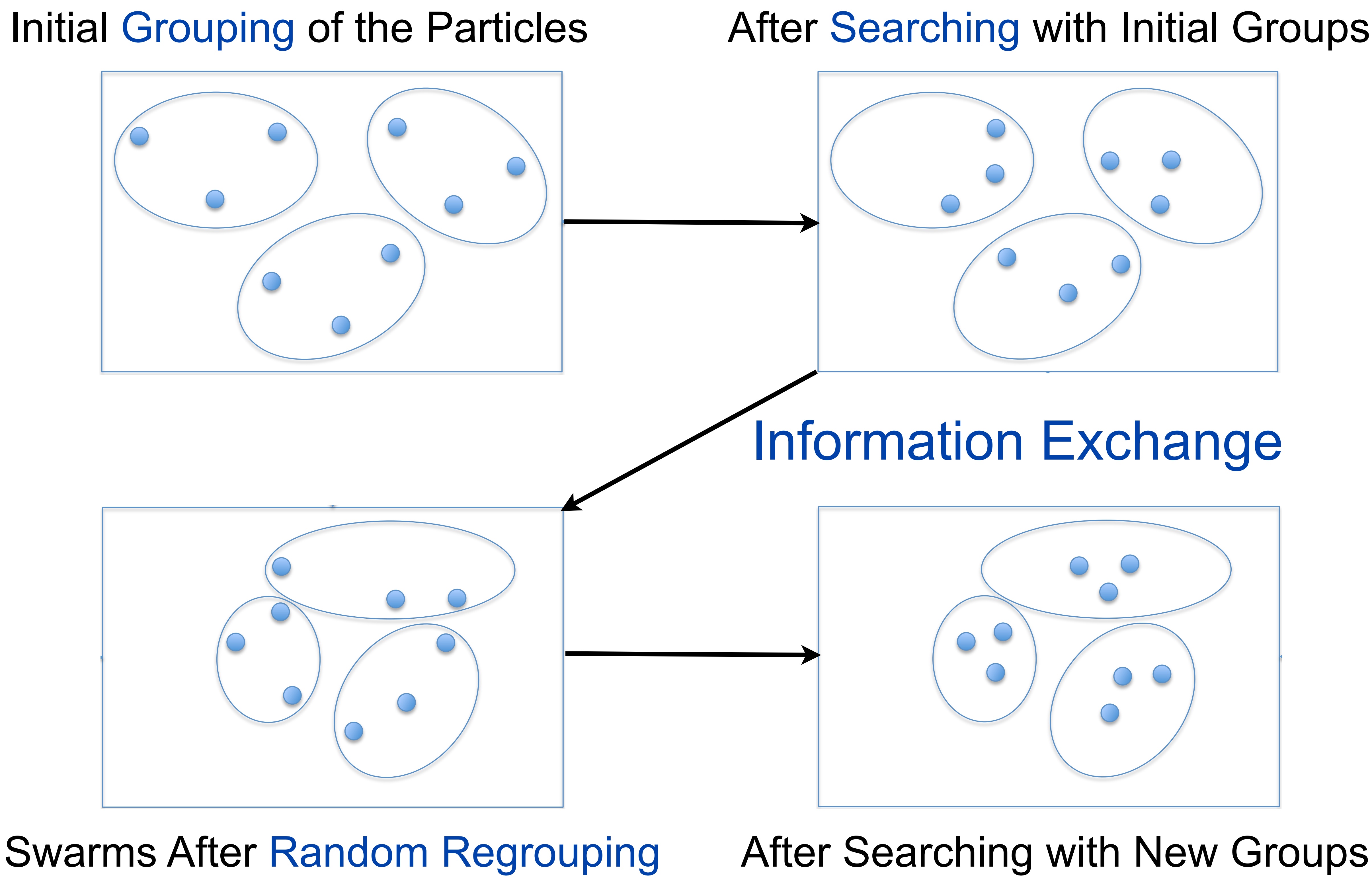
Local Neighborhood Based PSO

- Limitation of *gbest* PSO:
  - Premature convergence to local optima.
- Solution : *lbest* PSO
  - Best position achieved within the **neighborhood** influences a particle's velocity.

$$V_i(t+1) = \omega.V_i(t) + \phi_1.rand(0,1).(P_i - X_i(t)) + \phi_2.rand(0,1).(L_i - X_i(t))$$

  - L_i is best position achieved within the neighborhood.

Dynamic *lbest* PSO (D-LPSO)



How to Use D-LPSO for Learning Boolean Network?

Position Vector of a Particle:

$$v_{11}v_{12}..v_{1k}f_1 \mid v_{21}v_{22}..v_{2k}f_2 \mid \; . \; . \; . \; . \; . \; . \; \mid v_{n1}v_{n2}..v_{nk}f_n$$

Candidate Boolean Network

Genes are encoded in Binary

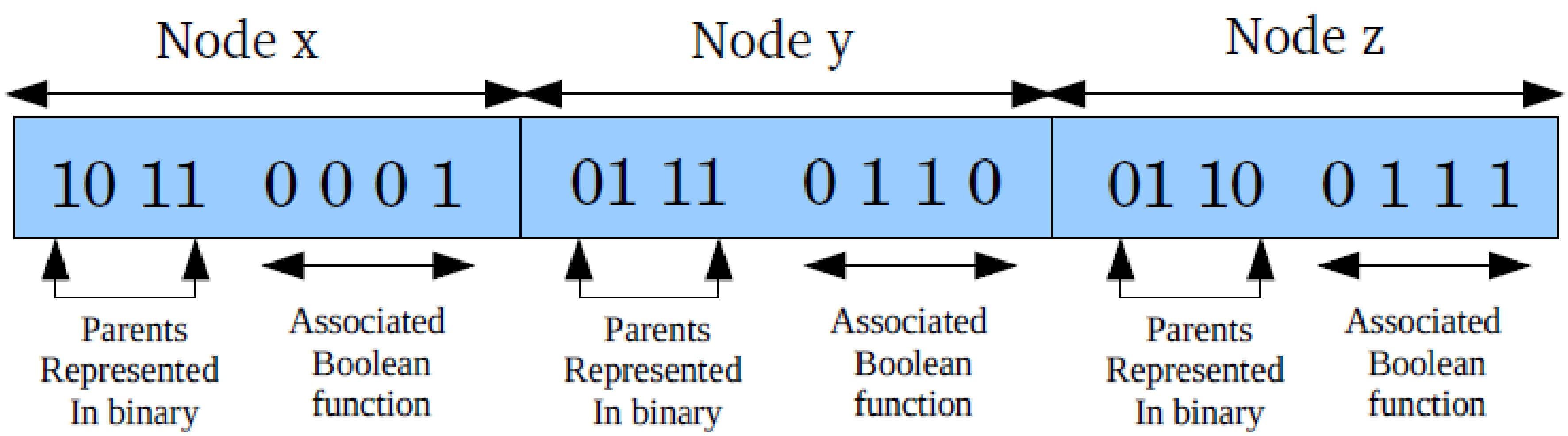
n genes

k inputs/gene

How to Use D-LPSO for Learning Boolean Network?

Position Vector of a Particle:

e.g: $n = 3, k = 2$



x : 01
y : 10
z : 11

x	y	z	x^*	y^*	z^*
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	0	1
1	1	0	0	1	1
1	1	1	1	0	1

How to handle Binary Data in PSO?

- Velocity of a particle gives the rate of change of bit's value.
- Two additional velocity vectors for each particle
- V_{i1} and V_{i0}, represents the change in velocity.
- Updated with the help of perturbations due to personal and social learning.

$$\text{If } pbest_i^d = 1 \text{ then } d_{i1,pbest}^d = c_1 r_1 (pbest_i^d - x_i^d) \text{ and } d_{i0,pbest}^d = -c_1 r_1 (pbest_i^d - x_i^d)$$

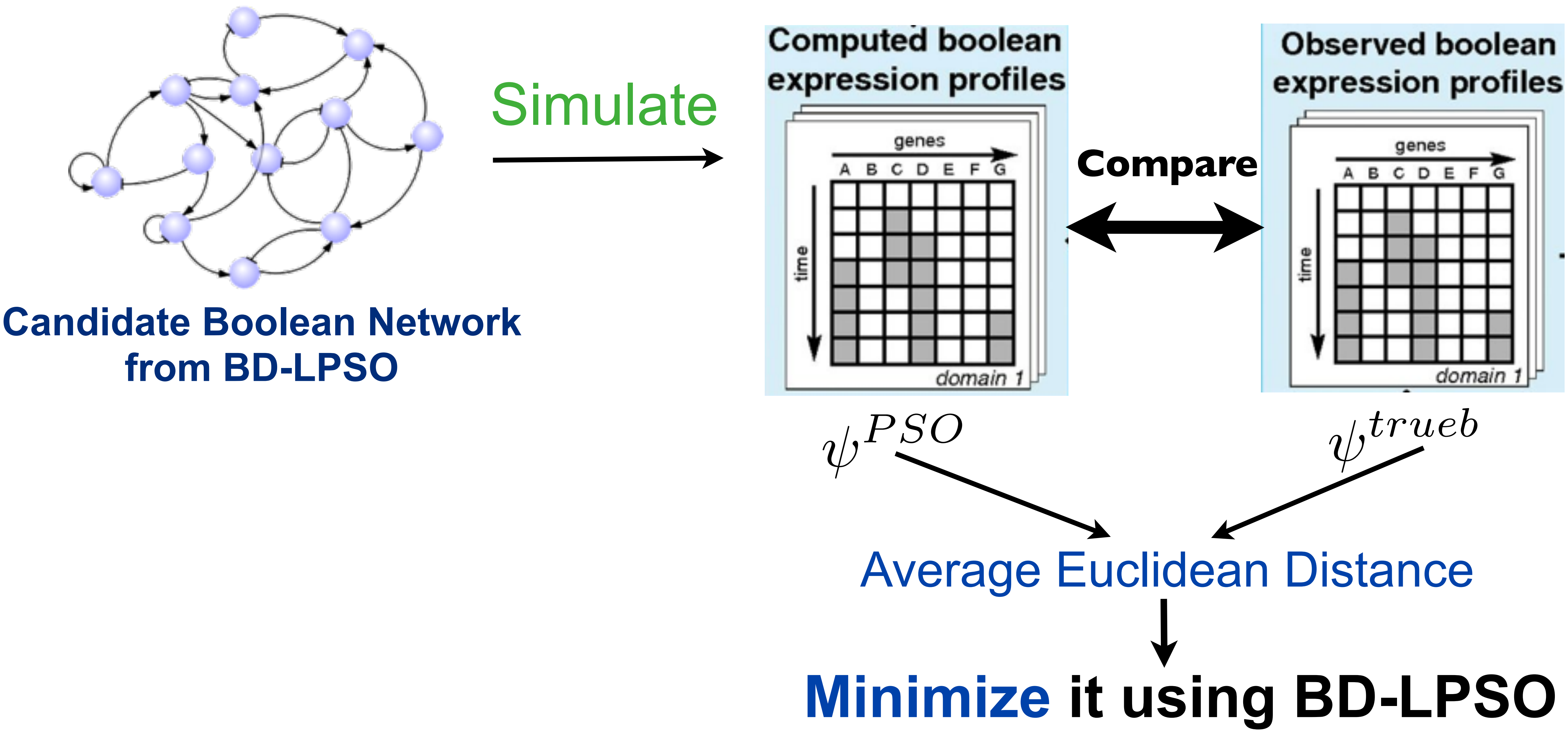
$$\text{If } pbest_i^d = 0 \text{ then } d_{i1,pbest}^d = -c_1 r_1 (pbest_i^d - x_i^d) \text{ and } d_{i0,pbest}^d = c_1 r_1 (pbest_i^d - x_i^d)$$

$$\text{If } lbest_i^d = 1 \text{ then } d_{i1,lbest}^d = c_2 r_2 (lbest_i^d - x_i^d) \text{ and } d_{i0,lbest}^d = -c_2 r_2 (lbest_i^d - x_i^d)$$

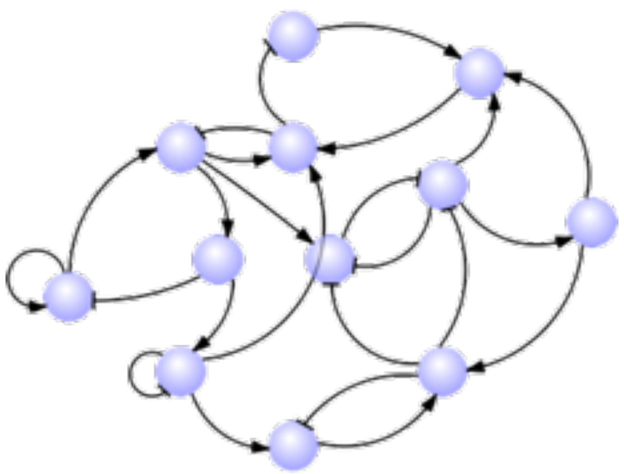
$$\text{If } lbest_i^d = 0 \text{ then } d_{i1,lbest}^d = -c_2 r_2 (lbest_i^d - x_i^d) \text{ and } d_{i0,lbest}^d = c_2 r_2 (lbest_i^d - x_i^d)$$

We call the resulting algorithm as **BD-LPSO**

Fitness Metric

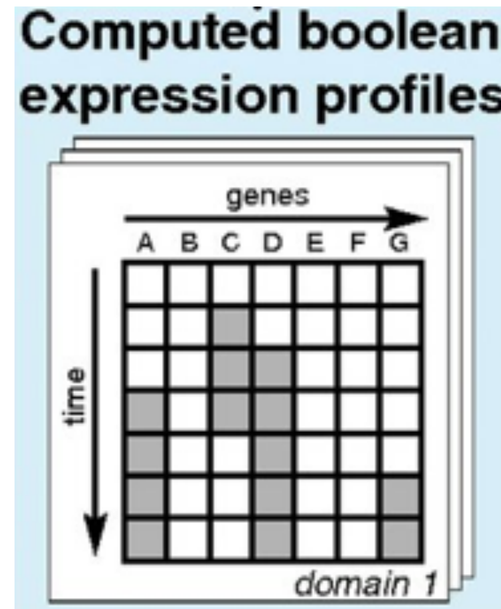


Fitness Metric

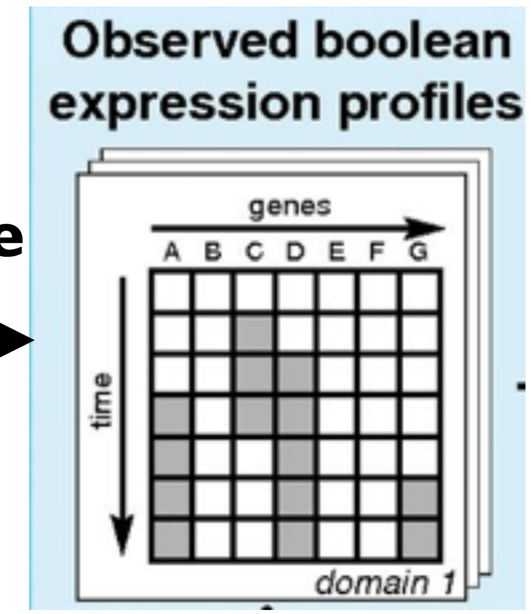


Candidate Boolean Network
from BD-LPSO

Simulate



Compare



ψ^{PSO} ψ^{trueb}

Average Euclidean Distance

Minimize it using BD-LPSO

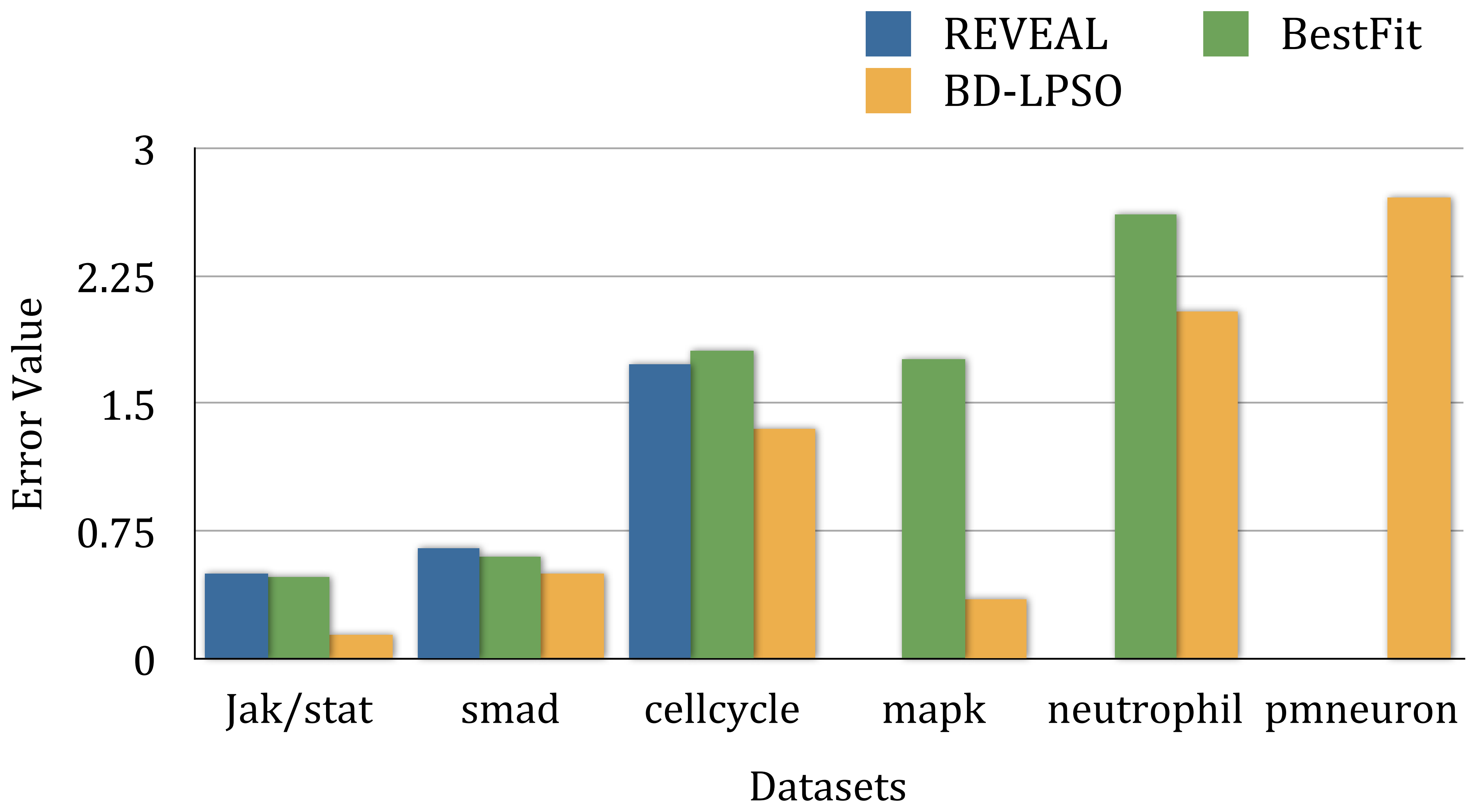
Simulation Scheme:

1. Synchronous
2. Asynchronous

Experimental Results

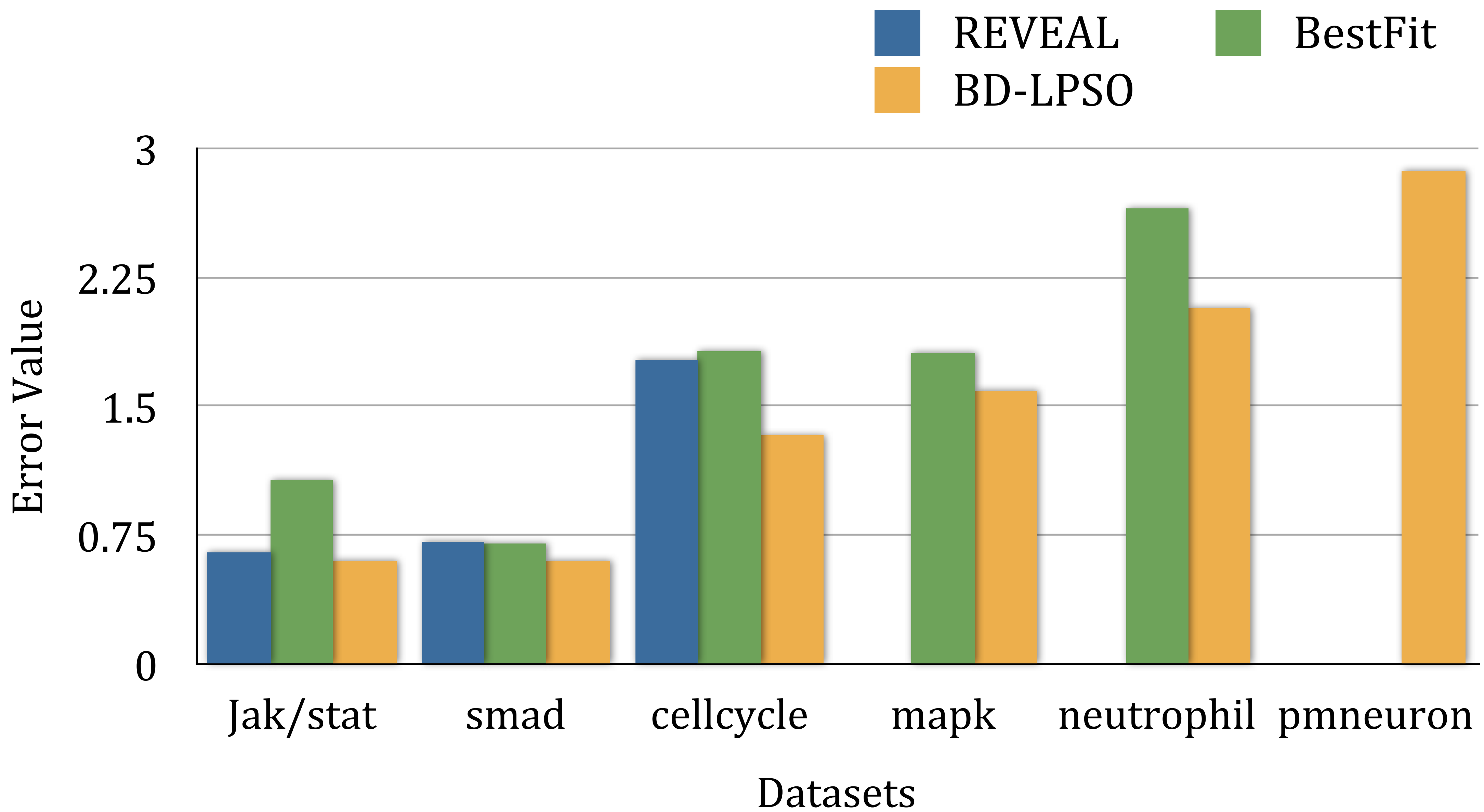
Error value is measured in terms of average Euclidean distance

Synchronous Simulation:

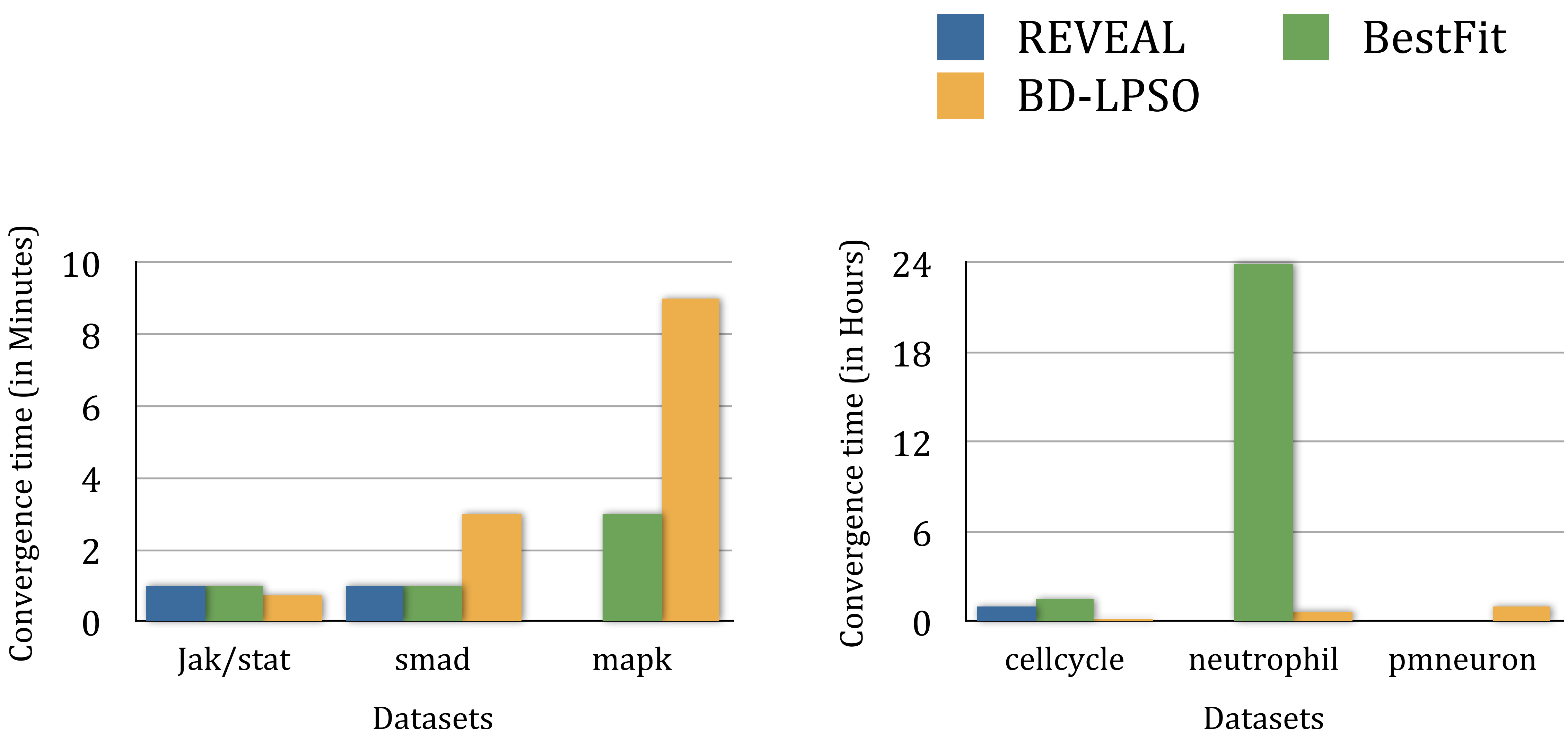


Experimental Results

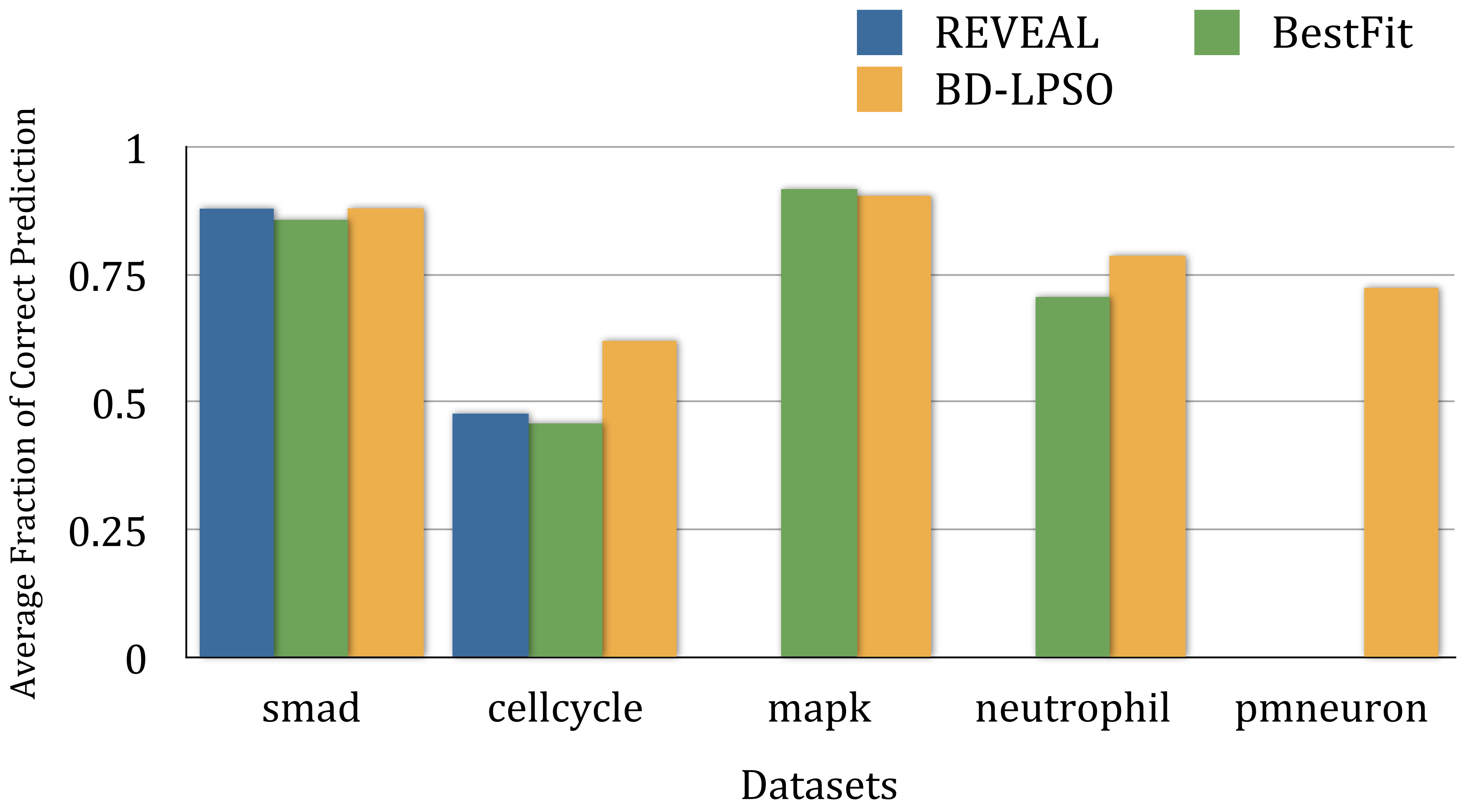
Asynchronous Simulation:



Comparison of Runtime

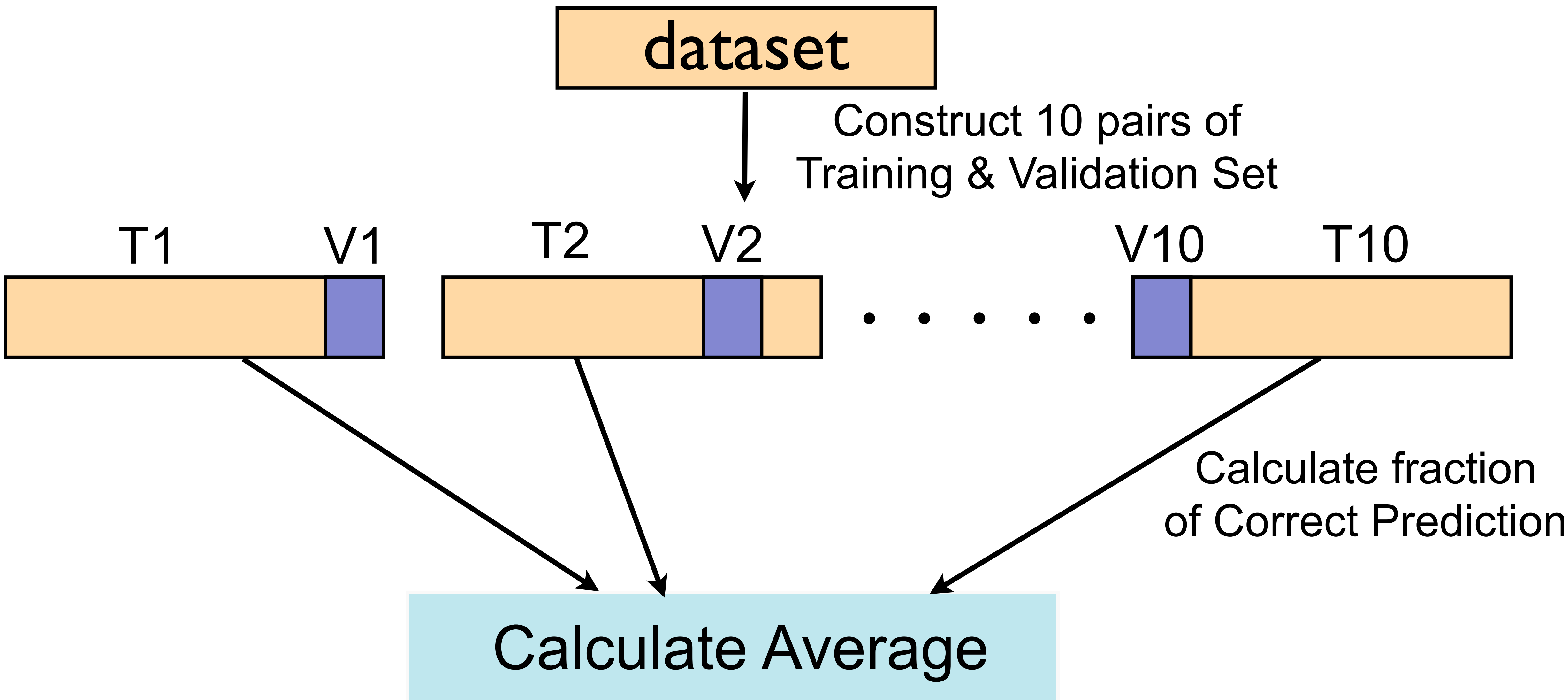


Comparison of Predictive Power



How to Calculate Predictive Power?

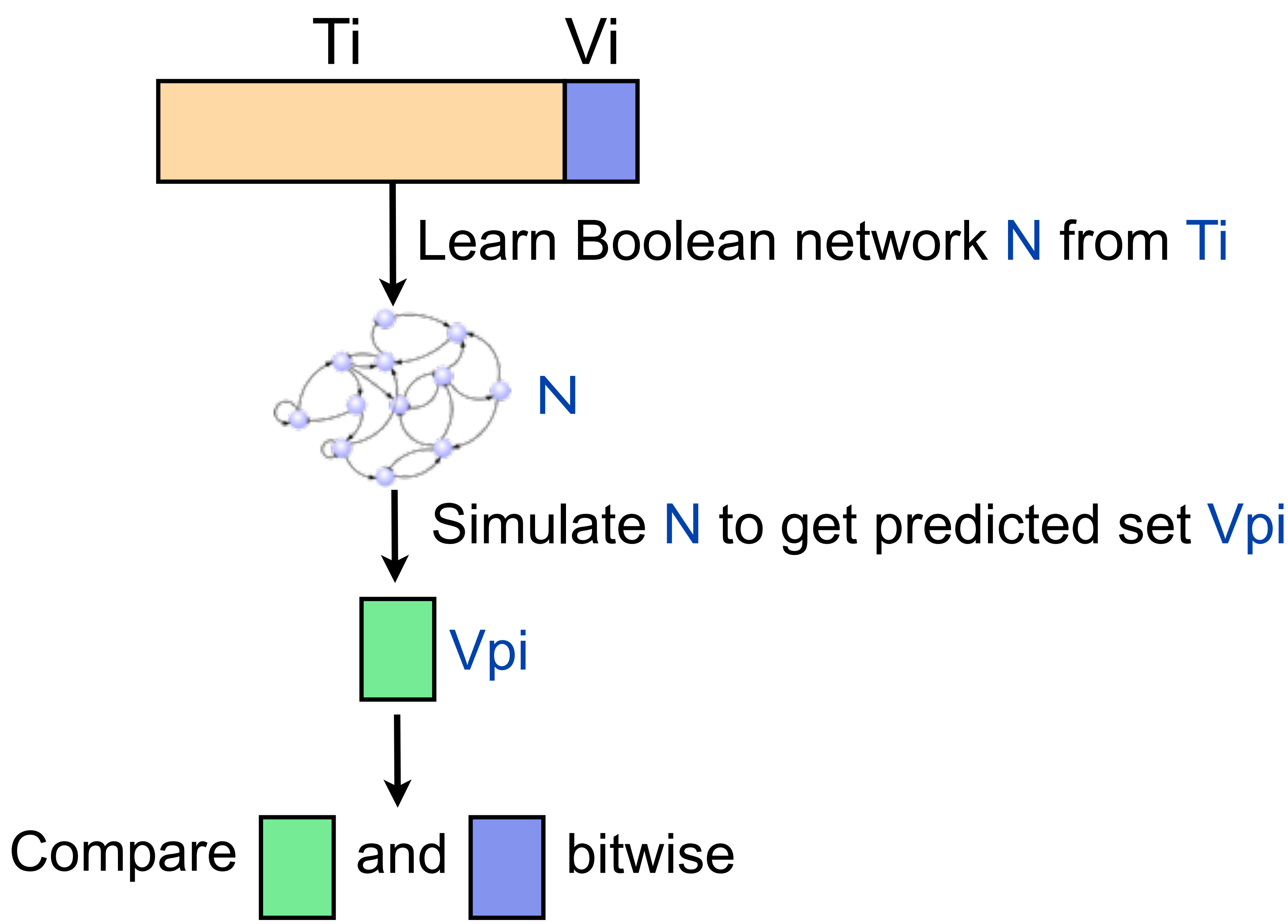
10 fold Cross Validation



Average over 10 runs

How to Calculate Predictive Power?

Fraction of Correct Prediction for a pair of Tr-Val Set :



Value is between 0 and 1

Future Work

- Integrated model of cellular network (signaling, metabolic and Gene Regulatory Networks).
- Learn more complex models (ODE). Supervised Learning using domain knowledge.
- Use of Gene knockout data along with time series data to infer the networks (two stage learning).

Conclusion

- BD-LPSO is an efficient algorithm for learning GRN from time series data.
- Better to learn BN in terms of
  - Accuracy
  - Predictive Power
  - Convergence Time.
- Specifically suitable for larger network with Complex dynamics.