COSC 348: Computing for Bioinformatics

Lecture 21
Gene Regulatory Networks 1 - Boolean networks

Lubica Benuskova

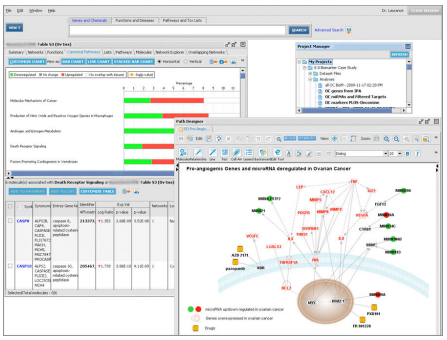
http://www.cs.otago.ac.nz/cosc348/

GRN inference versus modelling GRN dynamics

- GRN inference means a reconstruction of GRN based on experimental data.
- Modelling **the GRN dynamics** means simulating the changes of gene expression levels over time.
- For that we need a particular model:
  - **Boolean Networks**
  - Weight matrices
  - Bayesian Networks
  - Differential equations
  - other

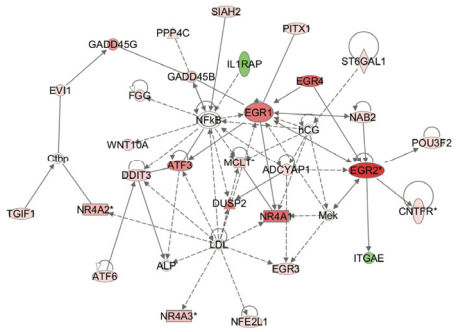
GRN inference by IPA (http://www.ingenuity.com/)

- Ingenuity® Pathway Analysis (IPA) software enables GRN inference based on the Ingenuity®Knowledge Base containing information on genes, proteins, drugs, etc. and molecular relationships.



GRN as a matrix of interactions

- Result of IPA software = matrix of interactions between genes. Genes are chosen e.g. based on selection applied to microarray data.



Modelling by Boolean network: assumptions

- Each gene is represented by a node in a *directed graph*, in which there is an arrow from one node/gene to another if and only if there is a causal link between the two nodes/genes.
- Each node in the graph can be in one of two states: **on** or **off**.
- "on" corresponds to the gene being expressed above some *threshold*; "off" corresponds to the gene being expressed below that threshold (i.e. gene expressions are binarised or discretised).
- Time is viewed as proceeding in discrete steps. At each step, the new state of a node is a *Boolean function* of the prior states of the nodes with arrows pointing towards it.

Boolean network model: formal equations

- Boolean network is a graph G(V, B), with a set of nodes (vertices) $V=\{v_i \mid i = 1,\ldots, n\}$, together with a set of Boolean functions $B = \{b_i \mid i=1,\ldots, n\}$, such that for $k \le n$

$$b_i : \{0,1\}^k \rightarrow \{0,1\}$$

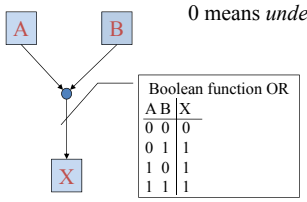
- Dynamics: The state of node v_i at time t is denoted as $x_i(t)$. Then, the state of that node at time t+1 is given by the transition formula:

$$x_i(t+1) = b_i\left(x_{i1}(t), x_{i2}(t),\ldots, x_{ik}(t)\right)$$

- where x_{ij} are the states of the nodes connected to v_i. These states can be 0 for underexpressed genes and 1 for overexpressed ones.

Example of simple Boolean network

1 means *overexpressed*
0 means *underexpressed*



Boolean function OR		
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

1) In this simple example, two genes (A and B) regulate gene X.
2) Their joint influence can be expressed by the logical function OR.
3) Expression levels of A and B at time t predict the expression level of X in the next time step (because their regulatory effect takes time)

Another example of Boolean network

a

c

Time *t*			Time *t* +1		
A	B	C	A'	B'	C'
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	1	1	1

b

A' = B
B' = A or C
C' = (A and B) or (B and C) or (A and C)

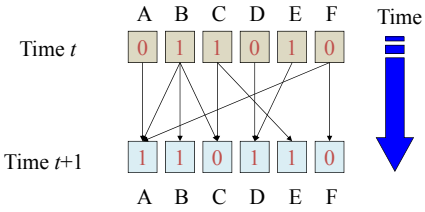
Figure 9. Target Boolean network for reverse engineering. Liang S, Fuhrman S, Somogyi R (1998) Pacific Symposium on Biocomputing 3:18-29.

- **(a)** Example of interaction network of three genes A,B,C
- **(b)** Boolean functions can be inferred from binarised gene expression data.
- **(c)** We need gene expression data over time to infer and verify these Boolean relationships.

Dynamics of Boolean network

Boolean network is a Graph consisting of G(V, B)

- V is a set of nodes (genes) with states $x_1, x_2, \ldots, x_n$
- B is a list of Boolean functions $b(x_1, x_2, \ldots, x_n)$



The gene expression levels change over time. These changes are governed by the set of Boolean functions.

Algorithm for updating the Boolean network

- Assuming we know the Boolean functions for each gene, i.e. we know the set $B = \{b_i \mid i=1,\ldots,n\}$, we start with initializing the values of gene expression in time 0, i.e.

$$x_i(t=0) = (1,0,1,0,0,\ldots,1)$$

- Then we proceed by updating the state of each gene in the next time step by applying the corresponding Boolean interactions:

$$x_i(t+1) = b_i(x_{i1}, x_{i2}, \ldots, x_{ik})$$

- We iterate until we reach a stable state, i.e. when for each gene *i* and for each time step *t*, it holds :

$$x_i(t) = x_i(t+1)$$

What to do if we have many gene expression data over time

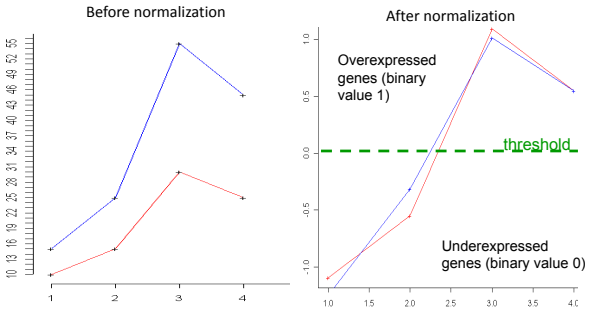
Time points

Genes

AffyID	Symbol	t0	t2	t4	t6	t8
1007_s_at	DDR1	105.265	63.89688	112.695	150.5448	86.05
1053_at	RFC2	30.19	18.05	31.185	47.8044	30.04
121_at	PAX8	238.915	143.2531	234.33	377.6472	219.735
1294_at	UBE1L	119.495	53.6125	110.175	144.7908	79.285
1316_at	THRA	30.19	18.05	31.185	47.8044	29.77
1431_at	CYP2E1	30.19	18.05	31.185	47.8044	29.385
1438_at	EPHB3	77.255	52.975	47.25	84.4116	49.955
1487_at	ESRRA	65.22	36.47188	78.625	115.542	73.31
1494_f_at	CYP2A6	58.23	30.71563	53.73	84.3612	55.515
1552256_a	SCARB1	116.085	58.225	120.74	167.9496	97.03
1552257_a	KIAA0153	75.455	38.14375	75.485	172.536	133.72
1552264_a	MAPK1	130.305	70.54375	161.525	219.0132	125.225
1552274_a	PXK	130.85	63.34375	62.34	56.826	31.9
1552275_s	PXK	131.01	56.99375	45.085	49.4508	30.3
1552277_a	MGC17337	139.87	123.425	325.305	538.104	321.925
1552279_a	MGC9564	86.975	44.50938	71.93	115.248	78.465
1552283_s	ZDHHC11	30.19	18.05	31.185	47.8044	29.385
1552287_s	AFG3L1	30.19	18.05	31.185	47.8044	29.385
1552291_a	FLJ20522	30.19	18.05	31.185	47.8044	29.385
1552295_a	SLC39A13	397.14	178.4313	378.815	613.7208	399.91

State vector $\mathbf{x}(t)$

Normalisation and binarisation of expression levels



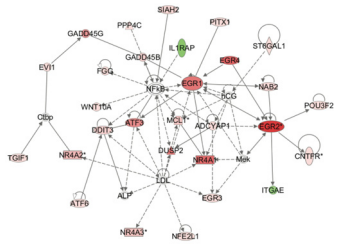
New "data": binarised gene expression data over time

- Overview:
  - $\mathbf{x}(t)$ is a vector representing binarised expression levels of N genes at time t
  - We want to build a model, i.e. set of Boolean functions for predicting $\mathbf{x}(t+1)$ given the previous series of states $\mathbf{x}(1),\ldots,\mathbf{x}(t)$

Gene ID	X(1)	X(2)	…	X(t)	X(t+1)
g_1	1	0	1	0	?
g_2	0	1	1	1	?
…	…	…	…	…	…
g_N	0	1	1	0	?

Use IPA to infer the relationships between genes

- Once we have set of interactions, then we generate Boolean functions randomly (while being constrained by existing interations and their directions).
- Then we iterate each GRN to see, which set of Boolean functions conforms with the changes in gene expression over time.



GRN behaves as a stable dynamic system

- *Trajectory* in time = series of state vector transitions.
- Example: $x_i(t+1) = b_i\left(x_{i1}(t), x_{i2}(t),\ldots, x_{ik}(t)\right)$
- If the dynamic system is stable then after certain time it will converge to a "repeating state" no matter what is its initial state.
- Based on the nature of this "repeating state" we distinguish
  - Fixed point attractor : a single state that repeats itself indefinitely
  - Limit cycle attractor: the system visits the same finite set of states periodically
  - Chaotic or strange attractor: we cannot predict the next state of the system, but the whole set of possible states is confined.

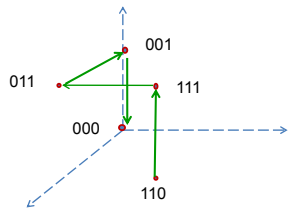
Asymptotic dynamics of Boolean network

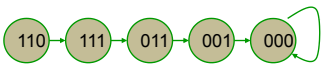
	t1	t2	t3	t4	t5	t6
Gene 1	1	1	0	0	0	0
Gene 2	1	1	1	0	0	0
Gene 3	0	1	1	1	0	0

Prediction!

- We have noticed that after time point t5, the state (i.e. vector of binarised gene expression levels) are always 0,0,0.
- Prediction: After some time, the dynamics reaches a stable state (in our example state 0,0,0), for which it holds: $x_i(t+1) = x_i(t)$

Trajectory in space

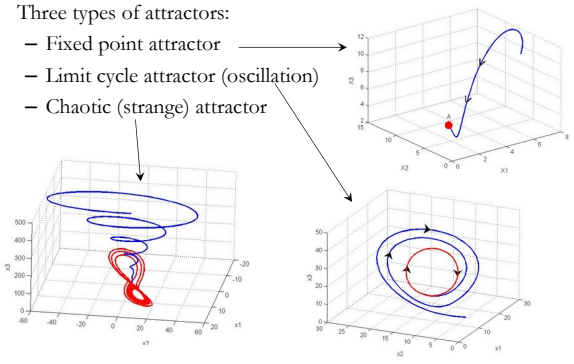




- After some time, the dynamics reaches a stable state (in our example state 0,0,0), for which it holds: $x_i(t+1) = x_i(t)$

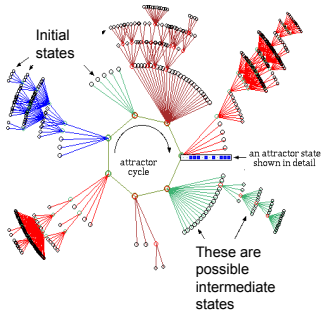
Three types of asymptotic attractors

- Three types of attractors:
  - Fixed point attractor
  - Limit cycle attractor (oscillation)
  - Chaotic (strange) attractor



http://www.scholarpedia.org/article/Attractor_network

Example of state space of Boolean network



- Different initial states undergo different transitions until they reach a stable state.
- Attractor states are stable under small perturbations
  - most perturbations cause the network to flow back to the attractor.
  - some genes are more important and changing their activation can cause the system to transit to a different attractor, i.e. cell state.

Summary of Boolean networks

- If the system is stable, then the dynamics of BN converges to an attractor, i.e. stable state(s) resistent to small perturbations.
- Converting real values of gene expression into '0' and '1' can hide important relationships, genes are not only on "OFF" or "ON" state.
- The biggest challenge is to infer Boolean functions, which is (1) often not possible, (2) interactions can change over time. Nevertheless, to infer these functions:
  - Use heuristics from biology to divide genes into small groups
  - Use clustering to divide genes into groups
  - Infer simple Boolean functions 'AND', 'OR', XOR', etc for small groups of genes or clusters of genes.
  - Join them together.
  - Or: generate many random GRNs and after iterating them, choose the best one