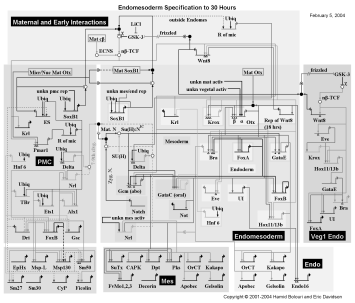
Computational Biology

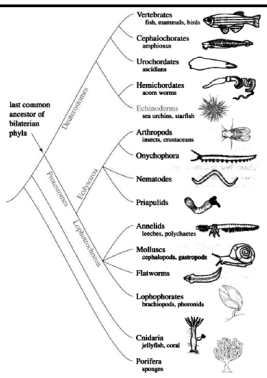
part V Modelling gene networks

Jaap Kaandorp

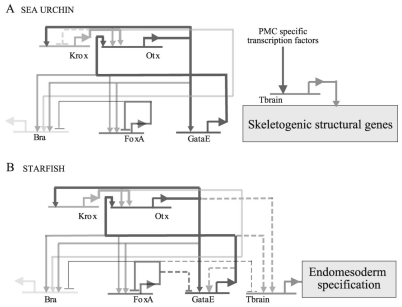
Genetic regulation in early development of a sea urchin



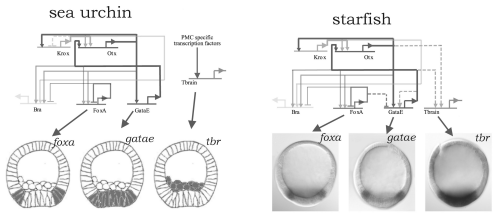
Metazoan phylogeny



Comparison of regulatory network architectures
(Hinman & Davidson, 2004)



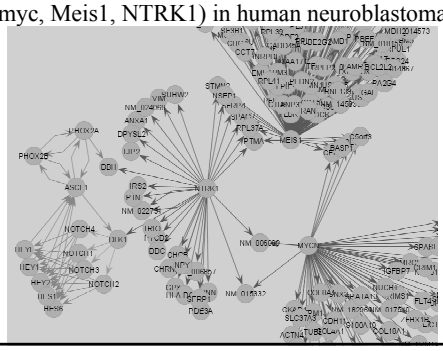
Evolutionary changes in expression patterns are predicted by alterations in upstream regulatory network architecture (Hinman & Davidson, 2004)



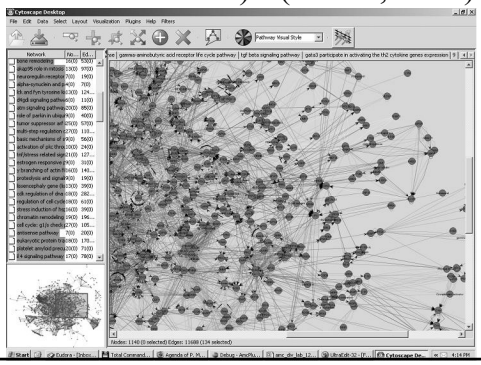
Example gene network

Experimentally identified
downstream pathways of three genes

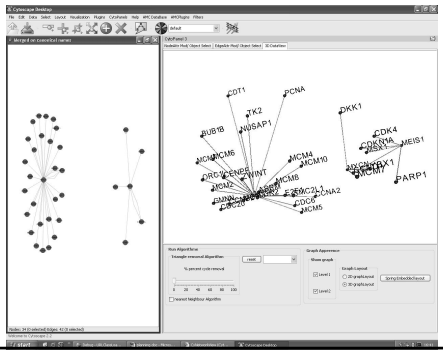
(N-myc, Meis1, NTRK1) in human neuroblastoma



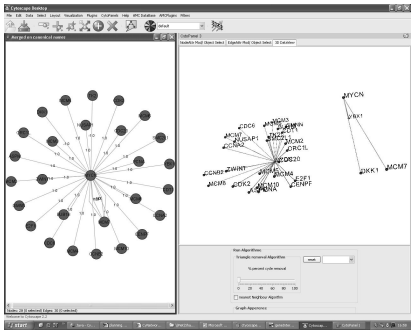
Example gene network (analysis of childhood cancer) I (Mahdi, 2006)



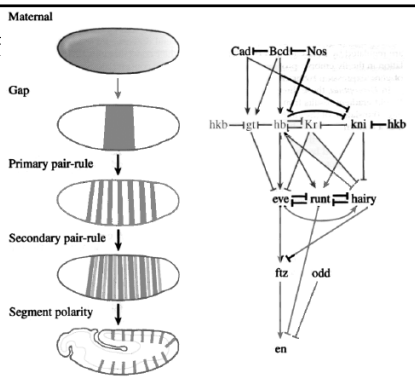
Example gene network (analysis of childhood cancer) II (Mahdi, 2006)



Example gene network (analysis of childhood cancer) III (Mahdi, 2006)



Development of morphogen gradients in time (left) and regulatory network for segment specification in *Drosophila* (Carroll,2001)



Possible components in a gene network model

- Forward flow of information from gene to mRNA to protein
- Positive and negative feedback loops
- Information exchange with metabolic pathways, signalling pathways

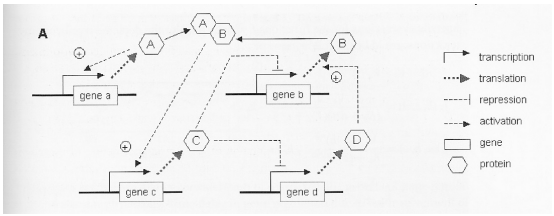
Different models for gene networks

- Boolean networks
- Bayesian networks
- Directed graphs
- Ordinary differential equations
- Partial differential equations
- Stochastic equations
- Rule-based formalisms

Genetic networks: the major challenges

- Understanding the dynamics and how gene networks regulate processes is a major (not solved in general) challenge! How do we model such a network? Complication: regulatory networks are very large, many details (needed in the models) are frequently missing
- major challenge: how to infer regulatory networks from gene expression data? How to infer model parameters?
- Major challenge: how do we couple models of regulatory networks and biomechanical models (for example models of growth and development)?

Modelling genetic networks, example



Modelling genetic networks with ODEs

$$\frac{dx_i}{dt} = f_i(x_1,...,x_n), i = 1,...,n$$

- Where x_i represent the concentrations of mRNAs, proteins, or other molecules and n number of genes

Modelling genetic networks with ODEs, example II

- Consider only mRNA quantities a,b,c and d:

$$\frac{da}{dt} = f_a(a)$$
$$\frac{db}{dt} = f_b(b,c,d)$$
$$\frac{dc}{dt} = f_a(a,b,c)$$
$$\frac{dd}{dt} = f_d(c,d)$$

Modelling genetic networks with ODEs, example III

- A possible model of the regulatory network describing quantities of mRNA:

$$\frac{da}{dt} = v_a - k_a \cdot a$$
$$\frac{db}{dt} = \frac{V_b \cdot d^{n_d}}{(K_b + d^{n_d})(K_{Ic} + c^{n_c})} - k_b \cdot b$$
$$\frac{dc}{dt} = \frac{V_c \cdot (a \cdot b)^{n_{ab}}}{K_c + (a \cdot b)^{n_{ab}}} - k_c \cdot c$$
$$\frac{dd}{dt} = \frac{V_d}{K_{Ic} + c^{n_c}} - k_d \cdot d$$

- Here k_a, k_b, k_c, k_d are rate constants of the degradation of a,b,c and d. v_a is a constant rate of expression

Modelling genetic networks with ODEs, example III

- A possible model of the regulatory network describing quantities of mRNA:

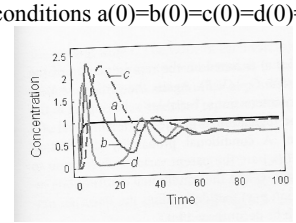
$$\frac{da}{dt} = v_a - k_a \cdot a$$
$$\frac{db}{dt} = \frac{V_b \cdot d^{n_d}}{(K_b + d^{n_d})(K_{Ic} + c^{n_c})} - k_b \cdot b$$
$$\frac{dc}{dt} = \frac{V_c \cdot (a \cdot b)^{n_{ab}}}{K_c + (a \cdot b)^{n_{ab}}} - k_c \cdot c$$
$$\frac{dd}{dt} = \frac{V_d}{K_{Ic} + c^{n_c}} - k_d \cdot d$$

$\frac{V_b \cdot d^{n_d}}{(K_b + d^{n_d})}$ Is Hill term with Max rate V_b, Dissociation constant K_b and Hill coefficient n_d

$(K_{Ic} + c^{n_c})$ Is inhibition term

Modelling genetic networks with ODEs, example IV

- Dynamics of the quantities of mRNA (v_a=1, k_a=1,V_b=1,K_b=5,K_{ic}=0.5, n_c=4;k_b=0.1,V_c=1,K_c=5;k_c=0.1,V_d=1, k_d=0,1; initial conditions a(0)=b(0)=c(0)=d(0)=0):



Modelling genetic networks with ODEs, example V

- A possible model of the regulatory network describing quantities of protein B:

$$\frac{dB}{dt} = D \cdot \frac{V_B \cdot b}{K_b + b} - k_B \cdot B - k_{AB} \cdot A \cdot B$$

$$\frac{db}{dt} = \frac{V_b}{(K_c + C^{n_c})} - k_b \cdot b$$

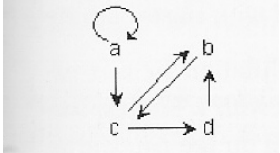
Modelling genetic networks with ODEs

- Advantage: can take into account detailed knowledge about regulatory network (including quantitative information; temporal information for example time delays, slow and fast processes; can be extended to PDE description including spatial information)
- Disadvantage: current lack of this detailed knowledge! Many parameters are not available, you need an additional method to estimate these parameters from actual data.
- Disadvantage: the ODE description is a macroscopic one, in many steps only a few molecules are involved.

Representation of gene networks as directed graphs

- A directed graph G is a tuple <V,E>, where V denotes a set of vertices and E a set of edges. The vertices i in V correspond to the genes (or other components of the system) and the edges correspond to their regulatory interactions
- In general edge tuple <i,j,*properties*>, where *properties* can be `+' activation or `-' inhibition, or a list of regulators and the influence on the edge

Representation of gene networks as a directed graphs, example



Representation of gene networks as directed graphs

- Many databases provide information about genetic regulation and are organized as annotated directed graphs (for example KEGG database)
- Directed graphs do not provide information about the dynamics of the network

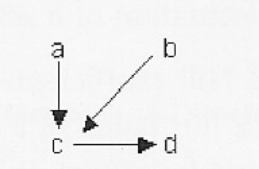
Representation of gene networks as directed graphs can be used to derive information about the network

- Tracing paths between genes
- Cycle in the network (feedback regulation)
- Comparison of gene regulatory networks
- The network complexity can be measured by connectivity

Baysian networks I

- A Baysian is based on the representation of the regulatory network as a directed acyclic graph G=<V,E>. Vertices i in V represent genes and edges denote regulatory interactions

Baysian networks, example II



Baysian networks III

- Variables x_i belonging to the vertices i denote a property relevant to the regulation, for example expression level of a gene or the amount of active protein
- A conditional probability distribution $p(x_i|L(x_i))$ is defined for each x_i, where $L(x_i)$ are the parent variables belonging to the direct regulators of i

Baysian networks IV

- The directed graph G and the conditional probability distributions together specify a joint probability distribution p(x) that determines the Baysian network. The joint probability distribution can be decomposed into:

$$p(x) = \prod_i p(x_i|L(x_i))$$

Baysian networks V

- Markov assumption: x_i is conditionally independent of its non-descendants given its parents
- Two graphs or Baysian networks are equivalents if they imply the same set of indepencies

Baysian networks VI, example

- Conditional independence relations

  $i(x_a; x_b), i(x_d; x_a, x_b | x_c)$

- Joint probability distribution

  $p(x_a, x_b, x_c, x_d) = p(x_a) \cdot p(x_b) \cdot p(x_c | x_a, x_b) \cdot p(x_d | x_c)$

Baysian networks VII

- Have been used to infer regulatory networks from gene expression data (find a class of networks that best explains the measured data)
- Problem: initial probability distribution function

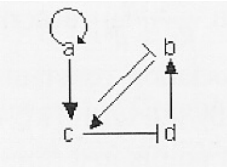
Boolean networks I

- In the Boolean network approach the expression level of each gene is assigned to a binary variable: a gene is considered to be either on (1) or off (0)
- The states of the genes are updated simultaneously in discrete time steps.
- New state can depend on previous state of the same gene or other genes

Boolean networks II

- N genes, N nodes of the network
- k interactions of a certain gene, k inputs node
- Every node 2 states, a network of N genes can be in 2^N different states
- State time t is N-dimensional vector
- State time t+1 depends on inputs, use Boolean rules, for k inputs, number of possible Boolean rules is 2^{2^k}

Boolean networks III, example



Boolean networks IV, example Boolean rules

$a(t+1) = a(t)$

$b(t+1) = (not(c(t)) and (d(t))$

$c(t+1) = (a(t)) and (b(t))$

$d(t+1) = (not(c(t))$

Boolean networks V, example, successive states

0000 → 0001	1000 → 1001
0001 → 0101	1001 → 1101
0010 → 0000	1010 → 1000
0011 → 0000	1011 → 1000
0100 → 0001	1100 → 1011
0101 → 0101	1101 → 1111
0110 → 0000	1110 → 1010
0111 → 0000	1111 → 1010

Boolean networks VI, trajectories and attractors

- The sequence of states leads to a trajectory
- Number of states is finite, number of possible transitions is finite
- Each trajectory leads either to a steady state or a state cycle. These states are called attractors. Transient states do not belong to an attractor. All states that lead to the same attractor constitute the basin of attraction

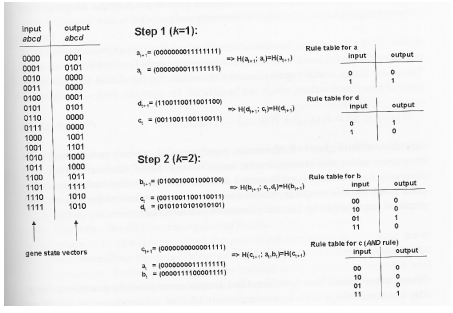
Boolean networks VII, reverse engineering (inferring networks)

- Can this be done from experimental data? In general not possible
- If we have time-dependent data and knockoout experiments, we might succeed. In time-dependent data: a strong expression of a transcription factor at time t will lead to activation or repression of gene expression of its targets at time t+1

Boolean networks VIII, reverse engineering (inferring networks), general strategy

- Consider a pair of consecutive time-dependent conditions (time t and t+1) on n gene probes
- Binarize the expression values and define a set of rules that allows the computation of binarized expression levels at time t+1 from those of time t

Boolean networks IX, reverse engineering (inferring networks), REVEAL algorithm (Liang et al.,1999)



Boolean networks IX, reverse engineering (inferring networks), REVEAL algorithm (Liang et al.,1999), example on blackboard

- 1. Identification of perfect input-output state pairs of connectivity k=1
- 2. Determination of the rules for the identified pairs at k=1
- 3. Identification of perfect input-output state pairs of connectivity k=2
- 4. Determination of the rules for the identified pairs at k=2
- 5. Identification of perfect input-output state pairs of connectivity k=p
- 6. Determination of the rules for the identified pairs at k=p
- 7. Stop if all genes have been assigned to rule, otherwise increment p and go to 5.

Boolean networks IX, reverse engineering (inferring networks), REVEAL algorithm (Liang et al.,1999), simulated data using the ODE model

- Expression data were generated with the ODE model
- Binarization of the data yielded 54 different state transitions, 21 are correct transitions
- Pre-processing state transitions after frequency of occurrence yielded a confidence level for each transition
- If only transitions highest confidence selected 6 out of 7 are correct.

Boolean networks IX, reverse engineering (inferring networks), REVEAL algorithm (Liang et al.,1999), simulated data using the ODE model

State transition	t	$t+1$	Confidence	Validity
1	0000	0001	1	True
2	1101	1111	0.5	True
3	1111	1110	0.625	False
4	1110	1010	0.5	True
5	1010	1000	0.3	True
6	1000	1001	1	True
7	1001	1101	0.5	True

Boolean networks IX, reverse engineering (inferring networks), REVEAL algorithm + simulated results from ODE model , conclusions

- Binarization is difficult and strongly influences result
- States are incomplete. In practice most of the state transitions are missing after binarization. Example only 7 of 15
- Availability of many time points is crucial. You need to filter correct states from false states, many state transitions are required (in example 100 time points)
- Time points not too close to each other. The selection of time points determines the granularity of the set of state transitions. Tradeoff between the detection of as many state transistions as possible and avoiding false-positive transitions.

Boolean networks IX, reverse engineering (inferring networks), general conclusions

- Advantage simple networks can be calculated quickly
- Disadvantage experimental data needs to be transformed into binary data and may lead to imperfect and ``noisy'' state vectors
- Currently Boolean networks are the only tractable method for inferring larger networks (nodes >100)