Modeling and simulation of gene-regulatory systems using Boolean networks
Gene-regulatory networks

- Gene expression is a regulatory process: Gene products influence the expression of other genes.
- Modeling and simulating gene-regulatory networks can provide deep insight into the functioning of cells.
- Simulations of gene-regulatory networks can replace costly biological experiments.
Boolean networks

- A gene is modeled as a binary variable ("expressed" or "not expressed")
- Each gene is associated with one or several Boolean transition functions with other genes as inputs
Synchronous Boolean networks

- each gene is associated with one transition function
- a state transition is performed by applying all transition functions synchronously
Asynchronous Boolean networks

- each gene is associated with one transition function
- a state transition is performed by applying a single randomly chosen transition function

\[ \begin{align*}
    x_1(t) & \quad \rightarrow \quad f_1 \\
    x_2(t) & \quad \rightarrow \quad f_2 \\
    x_3(t) & \quad \rightarrow \quad f_3
\end{align*} \]
Probabilistic Boolean networks

- a gene can have several transition functions, each of which is associated with a probability
- for a state transition, a random function is chosen for each gene, and the chosen functions are applied synchronously

\[ x_1(t) \rightarrow f_{11}(p=0.1) \rightarrow f_{12}(p=0.6) \rightarrow f_{13}(p=0.3) \rightarrow x_1(t+1) \]

\[ x_2(t) \rightarrow f_{21}(p=0.5) \rightarrow f_{22}(p=0.5) \rightarrow x_2(t+1) \]

\[ x_3(t) \rightarrow f_{31}(p=0.3) \rightarrow f_{32}(p=0.7) \rightarrow x_3(t+1) \]
Example: mammalian cell cycle network

\[ \text{CycD} = \text{CycD} \]
\[ \text{Rb} = (\neg\text{CycA} \land \neg\text{CycB} \land \neg\text{CycD} \land \neg\text{CycE}) \lor (p27 \land \neg\text{CycB} \land \neg\text{CycD}) \]
\[ \text{E2F} = (\neg Rb \land \neg\text{CycA} \land \neg\text{CycB}) \lor (p27 \land \neg Rb \land \neg\text{CycB}) \]
\[ \text{CycE} = (E2F \land \neg Rb) \]
\[ \text{CycA} = (E2F \land \neg Rb \land \neg\text{Cdc20} \land \neg(Cdh1 \land UbcH10)) \lor (\text{CycA} \land \neg Rb \land \neg\text{Cdc20} \land \neg(Cdh1 \land UbcH10)) \]
\[ p27 = (\neg\text{CycD} \land \neg\text{CycE} \land \neg\text{CycA} \land \neg\text{CycB}) \lor (p27 \land \neg(\text{CycE} \land \text{CycA}) \land \neg\text{CycB} \land \neg\text{CycD}) \]
\[ \text{Cdc20} = \text{CycB} \]
\[ \text{Cdh1} = (\neg\text{CycA} \land \neg\text{CycB}) \lor (\text{Cdc20}) \lor (p27 \land \neg\text{CycB}) \]
\[ \text{UbcH10} = \neg\text{Cdh1} \lor (\text{Cdh1} \land \text{UbcH10} \land (\text{Cdc20} \lor \text{CycA} \lor \text{CycB})) \]
\[ \text{CycB} = \neg\text{Cdc20} \land \neg\text{Cdh1} \]
Major tasks in network analysis (I)

Network construction

- formal modeling of statements in literature, e.g. “Gene A inhibits Gene B”
- inference of networks from time-series measurements of gene expression
- generation of random reference networks for comparison with real networks
Network simulation

- Identification of steady states and cycles (attractors)
- Identification of basins of attraction
- Markov chain simulations
- Artificial knock-out and overexpression experiments
The BoolNet package

- Supports construction and analysis of synchronous, asynchronous, and probabilistic Boolean networks.
- Includes the popular network reconstruction algorithms REVEAL and best-fit extension.
- Identification of synchronous and asynchronous attractors, Markov chain simulations.
- Simulation of knock-out and overexpression experiments.
- Visualization methods for gene dependencies, state transitions, attractors, basins of attraction etc.
- Random network generation and perturbation experiments.
BoolNet - Visualization of networks

```r
> plotNetworkWiring(cellcycle)
> m <- markovSimulation(examplePBN)
> plotPBNTransitions(m)
```
BoolNet - Visualization of attractors

> attr <- getAttractors(cellcycle)
> plotAttractors(attr, subset=1)
> plotStateGraph(attr)
Tutorial

- Learn how to build a Boolean network from "literature knowledge"
- Simulate the network
  - determine its attractors
  - examine reaction to changes (e.g. knock-out)
- Visualize the static and dynamic properties of the model