Neutral space analysis of Boolean regulatory networks

Gonzalo A. Ruz

joint work with Eric Goles

Complexity Research Center
Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez,
Santiago, Chile

Workshop, U. de Concepción, Chile
April 22, 2016

Outline

- Motivation
- Neutral space and neutral network
- Example: Fission yeast cell cycle Boolean network
- Evolutionary computation for neutral network construction
- Simulations
- Conclusion and research questions

Motivation (I)

- Reverse engineering $\rightarrow$ reconstruction (learning) of GRN models from experimental data (time-course gene expression).

Motivation (I)

- Reverse engineering $\rightarrow$ reconstruction (learning) of GRN models from experimental data (time-course gene expression).
- In practice, this is an *ill-posed* problem, number of time points of gene expression $<<$ number of genes analyzed.

Motivation (I)

- Reverse engineering $\rightarrow$ reconstruction (learning) of GRN models from experimental data (time-course gene expression).
- In practice, this is an *ill-posed* problem, number of time points of gene expression $<<$ number of genes analyzed.
- In Boolean networks, the available data is significantly less than the complete transition table.

Motivation (I)

- Reverse engineering $\rightarrow$ reconstruction (learning) of GRN models from experimental data (time-course gene expression).
- In practice, this is an *ill-posed* problem, number of time points of gene expression $<<$ number of genes analyzed.
- In Boolean networks, the available data is significantly less than the complete transition table.
- E.g., a network with 10 nodes $\rightarrow$ 1024 states, whereas typically the time-course gene expression data obtained from the lab is less than 3% (or even fewer) of all the configurations.

Motivation (I)

- Reverse engineering $\rightarrow$ reconstruction (learning) of GRN models from experimental data (time-course gene expression).
- In practice, this is an *ill-posed* problem, number of time points of gene expression $<<$ number of genes analyzed.
- In Boolean networks, the available data is significantly less than the complete transition table.
- E.g., a network with 10 nodes $\rightarrow$ 1024 states, whereas typically the time-course gene expression data obtained from the lab is less than 3% (or even fewer) of all the configurations.
- There may be several networks capable of representing/modeling the time-course gene expression.

Motivation (I)

- Reverse engineering $\rightarrow$ reconstruction (learning) of GRN models from experimental data (time-course gene expression).
- In practice, this is an *ill-posed* problem, number of time points of gene expression $<<$ number of genes analyzed.
- In Boolean networks, the available data is significantly less than the complete transition table.
- E.g., a network with 10 nodes $\rightarrow$ 1024 states, whereas typically the time-course gene expression data obtained from the lab is less than 3% (or even fewer) of all the configurations.
- There may be several networks capable of representing/modeling the time-course gene expression.
- **Which of those networks is the most plausible?**

Motivation (II)

- Something similar also occurs, when given a Boolean network that models a certain biological phenomenon.

Motivation (II)

- Something similar also occurs, when given a Boolean network that models a certain biological phenomenon.
- Typically the biological properties of the model is captured only by a small subset of states of the total 2^n configurations.

Motivation (II)

- Something similar also occurs, when given a Boolean network that models a certain biological phenomenon.
- Typically the biological properties of the model is captured only by a small subset of states of the total 2^n configurations.
- This opens the opportunity to search for other synthetic networks that have the same function as the original *base* model, also known as the *wildtype network*, but not necessarily all the dynamics.

Motivation (II)

- Something similar also occurs, when given a Boolean network that models a certain biological phenomenon.
- Typically the biological properties of the model is captured only by a small subset of states of the total 2^n configurations.
- This opens the opportunity to search for other synthetic networks that have the same function as the original *base* model, also known as the *wildtype network*, but not necessarily all the dynamics.

$\Rightarrow$ Study the neutral space of a regulatory network model.

Neutral space

The study of the neutral space consists of analyzing topological and dynamical properties of different regulatory networks that share the same function.

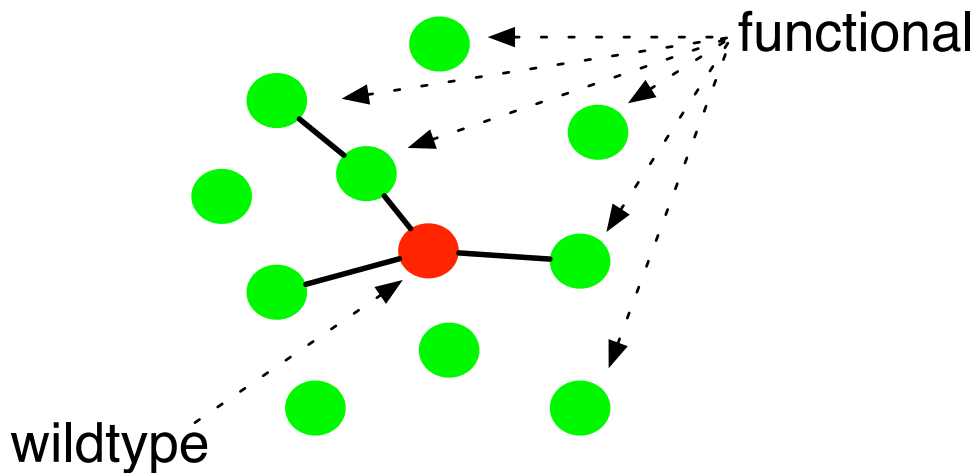
Neutral space

The study of the neutral space consists of analyzing topological and dynamical properties of different regulatory networks that share the same function.

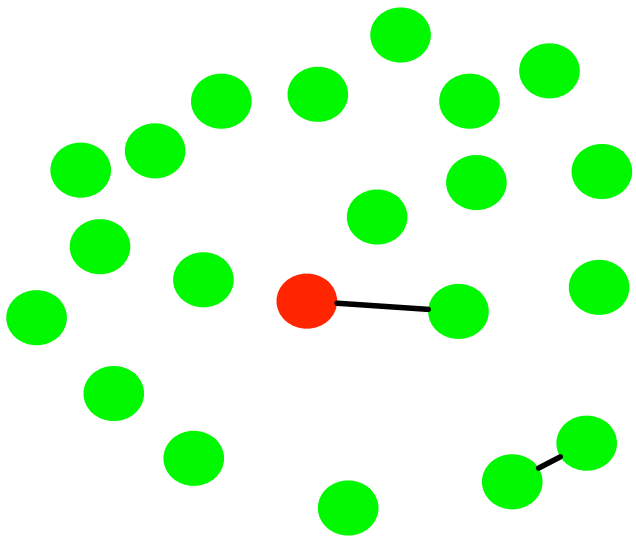
$\Rightarrow$ This analysis is carried out through the construction of a *neutral network*.

Neutral network

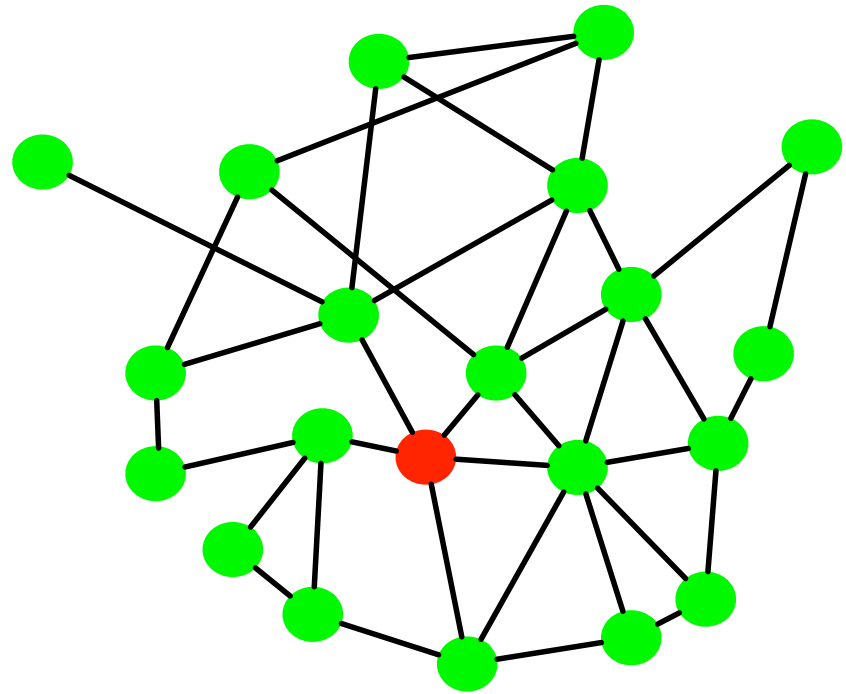
- Neutral network: network of networks
- Each node represents a network
- Two nodes connected means that the Hamming distance between the interaction (adjacency) matrix of one network and the other is one.



Robustness in neutral networks

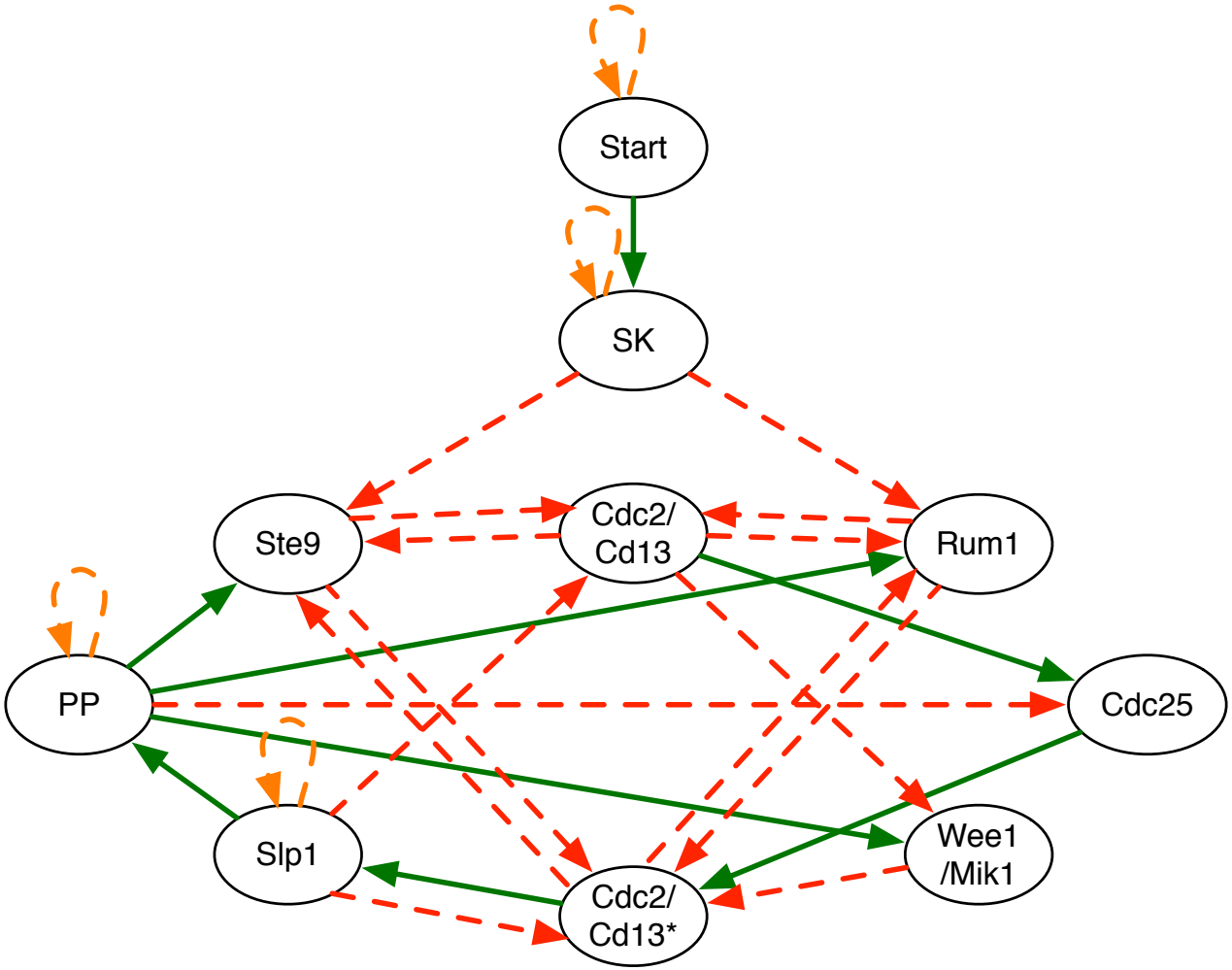


(a) Low robustness



(b) High robustness

E.g. The fission yeast cell-cycle Boolean network



M. I. Davidich and S. Bornholdt, Boolean network model predicts cell cycle sequence of fission yeast, *PLoS ONE*, vol. 3(2), p. e1672, 2008.

Updating the nodes values

$$
\begin{aligned}
x_i(t+1) &= u\left(\sum_{j=1}^{n} w_{ij}x_j - \theta_i\right) \\
&= \begin{cases} 0, & \text{if } \sum_{j=1}^{n} w_{ij}x_j - \theta_i < 0 \\ 1, & \text{if } \sum_{j=1}^{n} w_{ij}x_j - \theta_i > 0 \\ x_i(t), & \text{if } \sum_{j=1}^{n} w_{ij}x_j - \theta_i = 0 \end{cases}
\end{aligned}
$$

$W =$

	Start	*SK*	*Cdc2/Cdc13*	*Ste9*	*Rum1*	*Slp1*	*Cdc2/Cd13**	*Wee1/Mik1*	*Cdc25*	*PP*
Start	−1	0	0	0	0	0	0	0	0	0
SK	1	−1	0	0	0	0	0	0	0	0
Cdc2/Cdc13	0	0	0	−1	−1	−1	0	0	0	0
Ste9	0	−1	−1	0	0	0	−1	0	0	1
Rum1	0	−1	−1	0	0	0	−1	0	0	1
Slp1	0	0	0	0	0	−1	1	0	0	0
*Cdc2/Cd13**	0	0	0	−1	−1	−1	0	−1	1	0
Wee1/Mik1	0	0	−1	0	0	0	0	0	0	1
Cdc25	0	0	1	0	0	0	0	0	0	−1
PP	0	0	0	0	0	1	0	0	0	−1

$\Theta = (0 \quad 0 \quad -0.5 \quad 0 \quad 0 \quad 0 \quad 0.5 \quad 0 \quad 0 \quad 0)^T$

Temporal evolution of sate vectors defining the fission yeast cell cycle

Time	Start	SK	Cdc2/Cdc13	Ste9	Rum1	Slp1	Cdc2/Cd13*	Wee1/Mik1	Cdc25	PP	Phase
1	1	0	0	1	1	0	0	1	0	0	START
2	0	1	0	1	1	0	0	1	0	0	G_1
3	0	0	0	0	0	0	0	1	0	0	G_1/S
4	0	0	1	0	0	0	0	1	0	0	G_2
5	0	0	1	0	0	0	0	0	1	0	G_2
6	0	0	1	0	0	0	1	0	1	0	G_2/M
7	0	0	1	0	0	1	1	0	1	0	G_2/M
8	0	0	0	0	0	1	0	0	1	1	M
9	0	0	0	1	1	0	0	1	0	1	M
10	0	0	0	1	1	0	0	1	0	0	G_1

Problem description

- We desire to construct a neutral network of the fission yeast cell cycle network.

Problem description

- We desire to construct a neutral network of the fission yeast cell cycle network.
- Regulatory networks that reproduce the ten state sequences of the fission yeast cell cycle will be called *functional* networks, while the original fission yeast cell cycle network will be called *wildtype* network.

Problem description

- We desire to construct a neutral network of the fission yeast cell cycle network.
- Regulatory networks that reproduce the ten state sequences of the fission yeast cell cycle will be called *functional* networks, while the original fission yeast cell cycle network will be called *wildtype* network.
- The search space for functional networks is huge, which makes it a difficult problem.

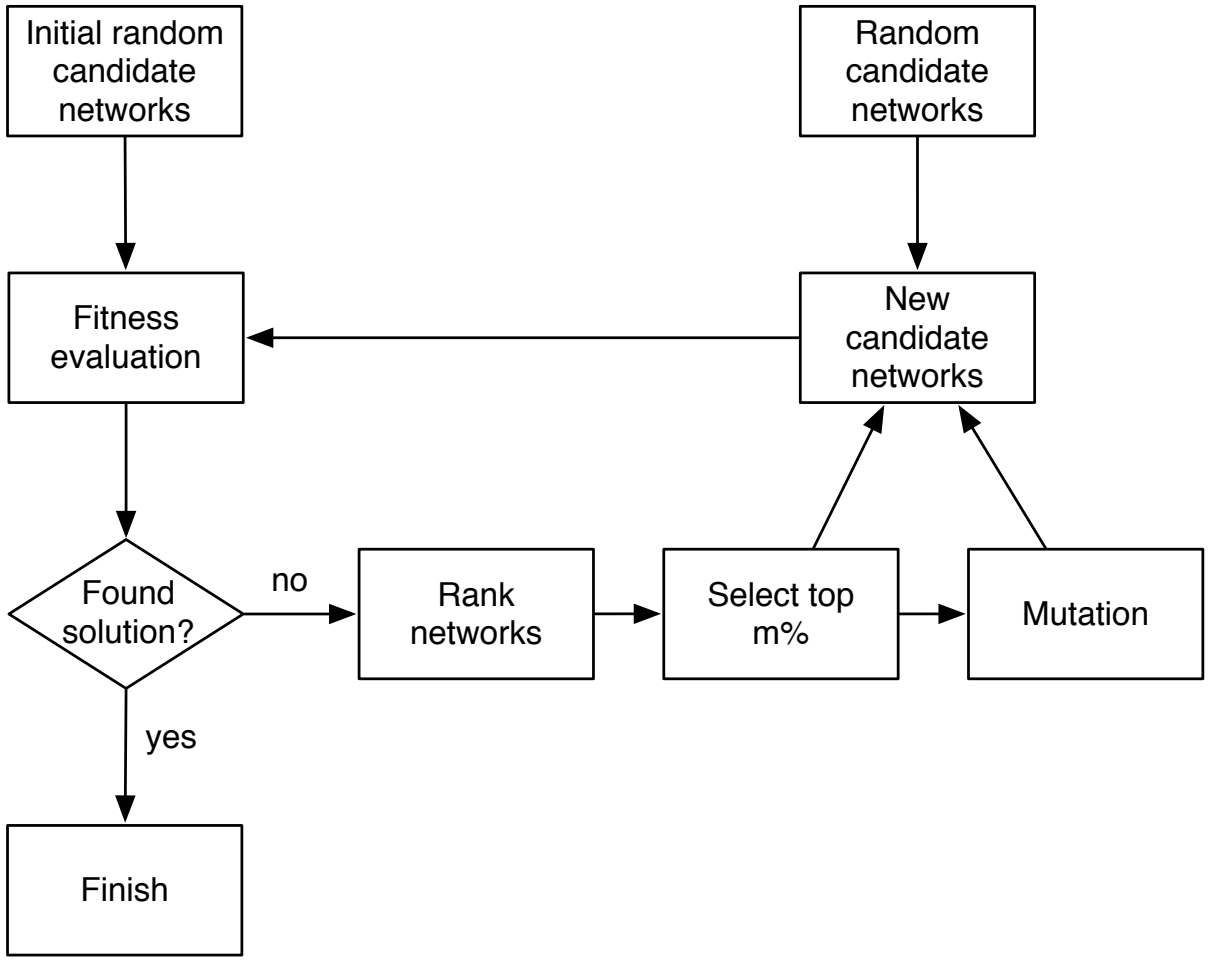
Problem description

- We desire to construct a neutral network of the fission yeast cell cycle network.
- Regulatory networks that reproduce the ten state sequences of the fission yeast cell cycle will be called *functional* networks, while the original fission yeast cell cycle network will be called *wildtype* network.
- The search space for functional networks is huge, which makes it a difficult problem.
- The search consists in finding the weight matrix elements w_{ij} and the threshold vector elements θ_i that can replicate the desired state sequences.

Problem description

- We desire to construct a neutral network of the fission yeast cell cycle network.
- Regulatory networks that reproduce the ten state sequences of the fission yeast cell cycle will be called *functional* networks, while the original fission yeast cell cycle network will be called *wildtype* network.
- The search space for functional networks is huge, which makes it a difficult problem.
- The search consists in finding the weight matrix elements w_{ij} and the threshold vector elements θ_i that can replicate the desired state sequences.
- An opportunity for intelligent search strategies arises, in particular the use of evolutionary computation.

Evolution strategy proposed to search for functional networks



The wildtype weight matrix is changed using the following rule:
Rule 1

1. Select randomly a position (i, j) in the matrix.
2. If the position contains a non-zero number, then replace by a zero.
3. Else, replace with a value selected randomly from the following set $\{-2, -1, 1, 2\}$.

The wildtype weight matrix is changed using the following rule:
Rule 1

1. Select randomly a position (i, j) in the matrix.
2. If the position contains a non-zero number, then replace by a zero.
3. Else, replace with a value selected randomly from the following set $\{-2, -1, 1, 2\}$.

The wildtype threshold vector is changed using the following rule:
Rule 2

1. Select randomly a position i in the vector.
2. Replace with a value selected randomly from the following set $\{-2, -1, -1/2, 0, 1/2, 1, 2\}$.

both rules are repeated *ngh* times, where *ngh* is selected randomly in the range of $[1, 30]$, for every new candidate network generated.

Fitness evaluation

The fitness function for the Boolean regulatory network B, is computed by the deviation of the network's output, defined by o_i for each node i, and the target value s_i (sequence of the cell cycle) for each node i:

$$fitness(B) = \frac{1}{10n} \sum_{t=1}^{10} \sum_{i=1}^{n} (o_i(t) - s_i(t))^2 \quad (1)$$

where n is the number of nodes in the network, and 10 is the number of state vector sequences that the network must contain.

Mutation

New candidate networks are generated using the following rule:

Rule 3

1. Select randomly one of the top $m\%$ solutions.
2. Mutate the selected solution. This is done by applying Rule1 and Rule2 with $ngh = 1$.

Simulations

- Simulation 1: ES was set to search for 10000 functional networks.

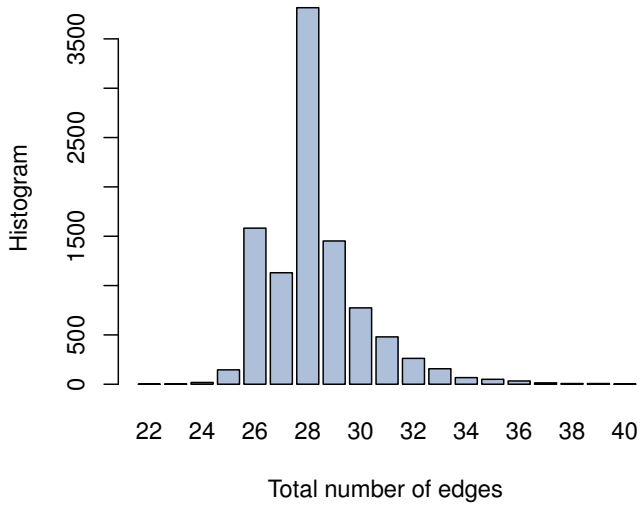
Simulations

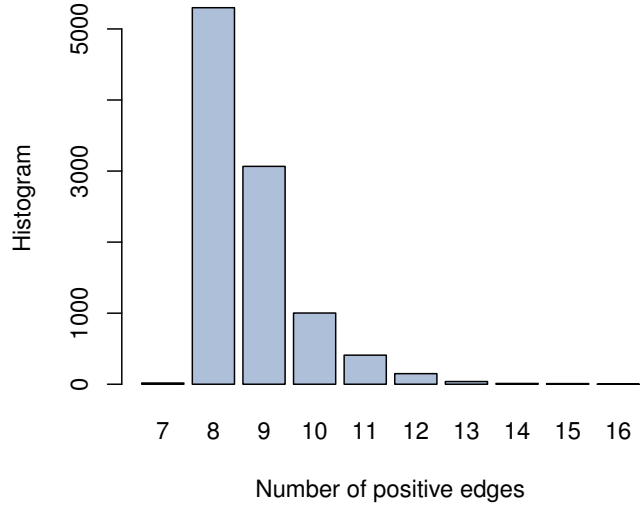
- Simulation 1: ES was set to search for 10000 functional networks.
- Simulation 2: we search for the connected component of the wildtype network. To do this, we run the ES but using $ngh = 1$ for the initial random candidate network stage.

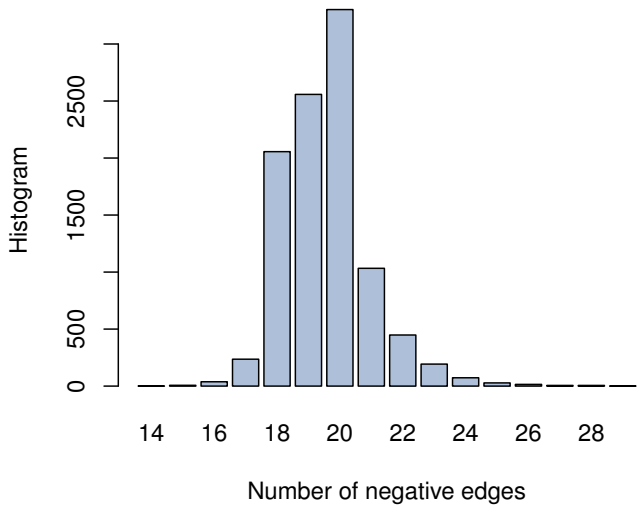
Simulations

- Simulation 1: ES was set to search for 10000 functional networks.
- Simulation 2: we search for the connected component of the wildtype network. To do this, we run the ES but using $ngh = 1$ for the initial random candidate network stage.
- Simulation 3: Results using a standard real-valued GA.

Histograms of the functional networks topologies of the neutral graph

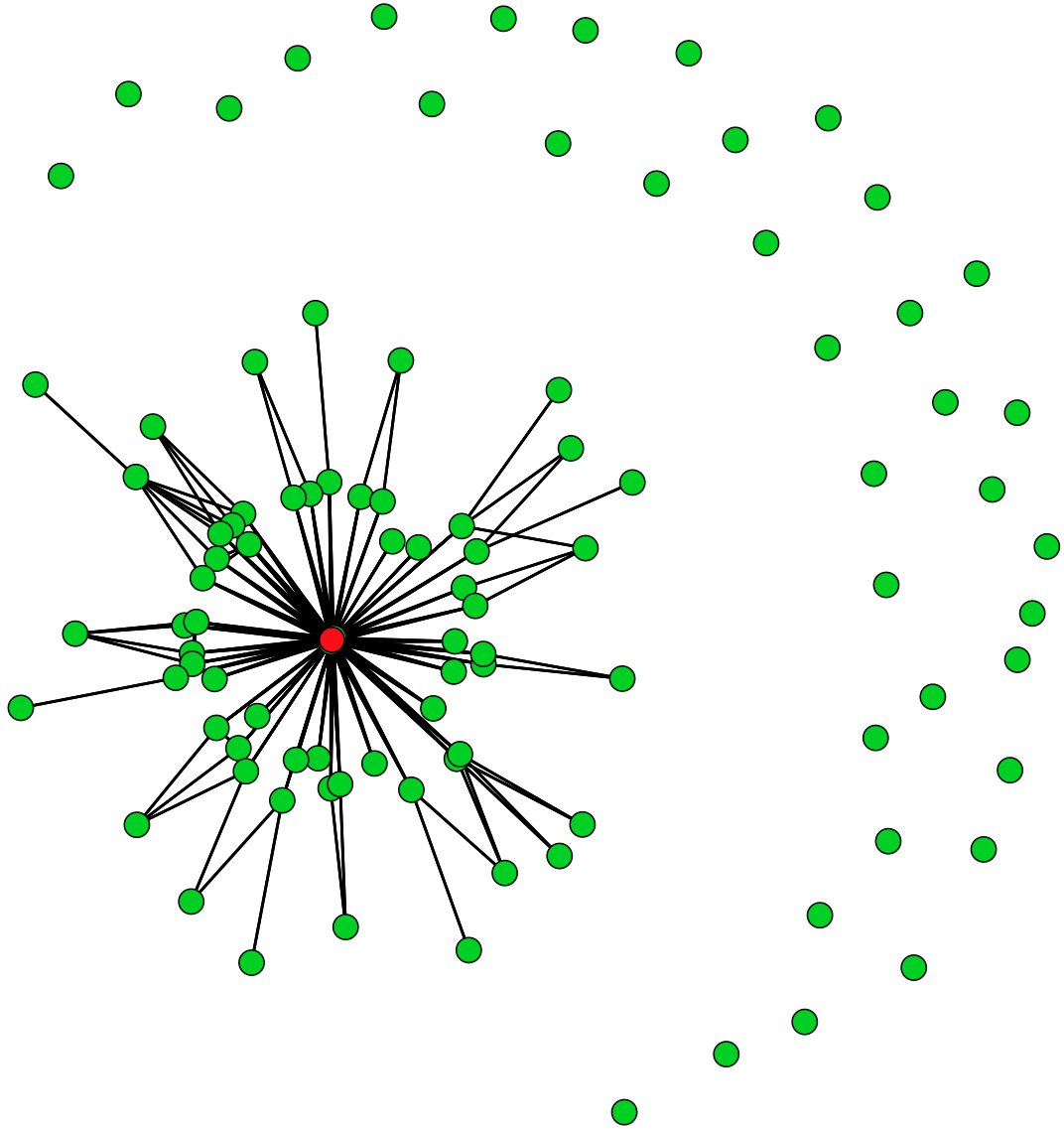




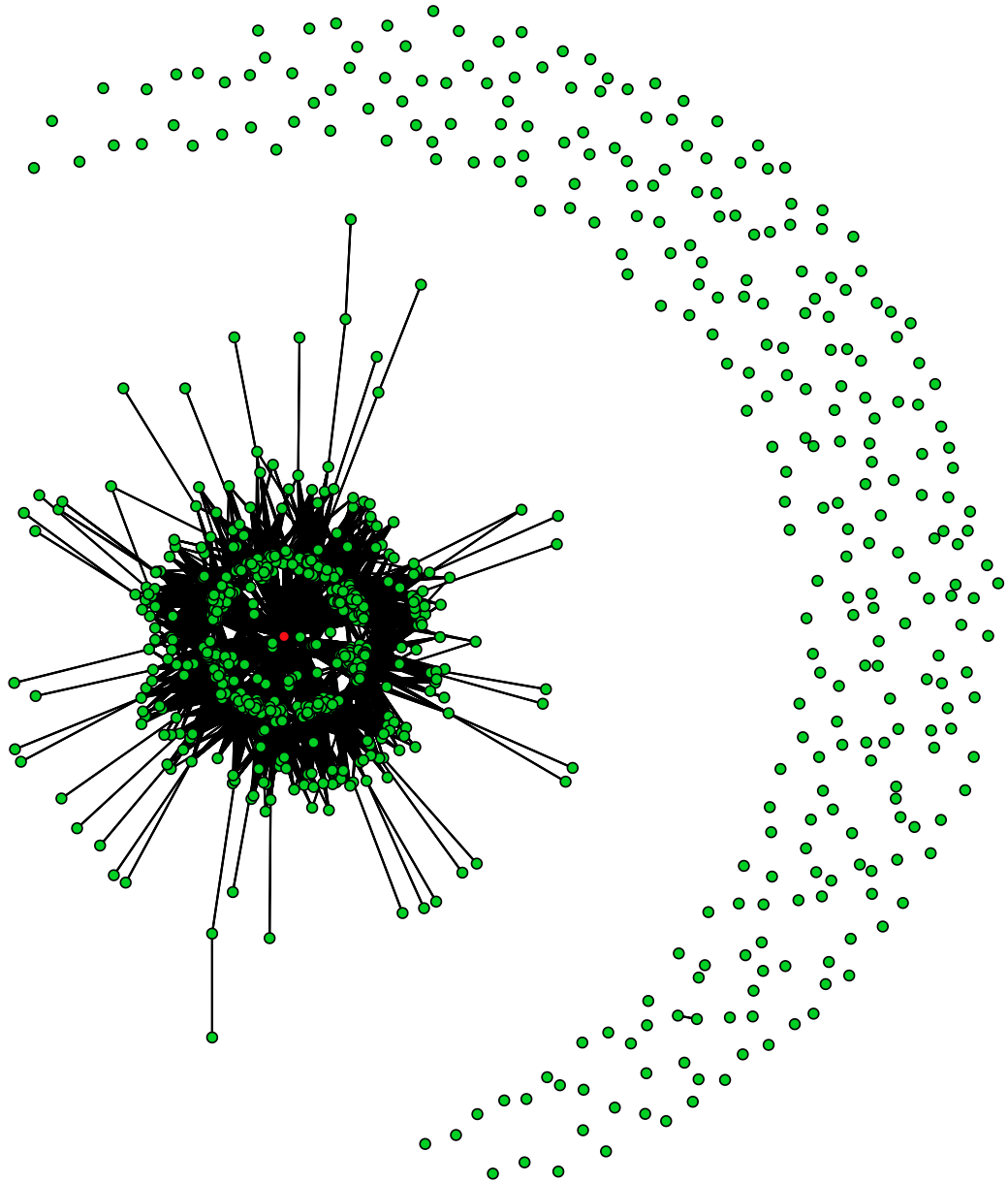


Wildtype network
total edges: 27
positive edges:8
negative edges:19

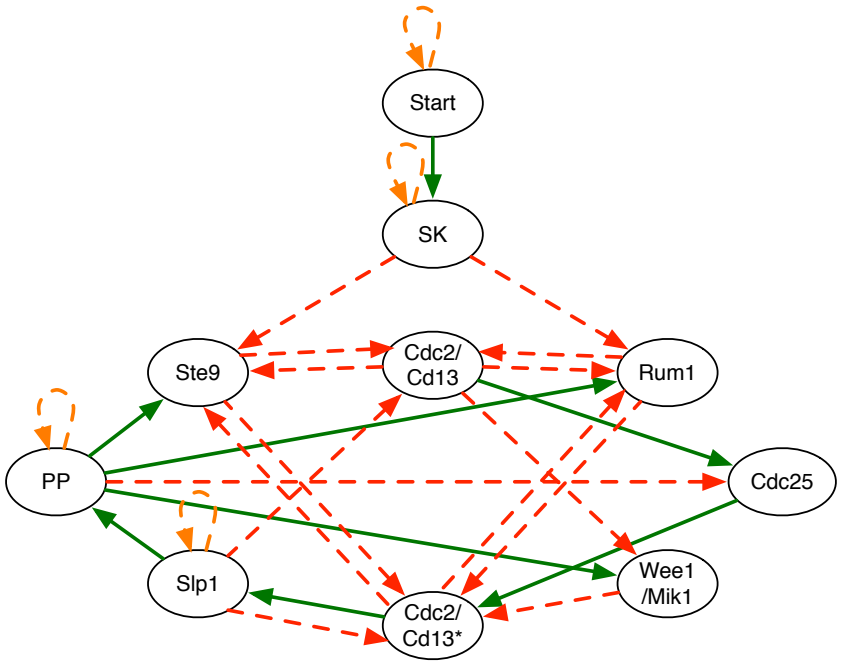
Neutral graph using 100 functional networks

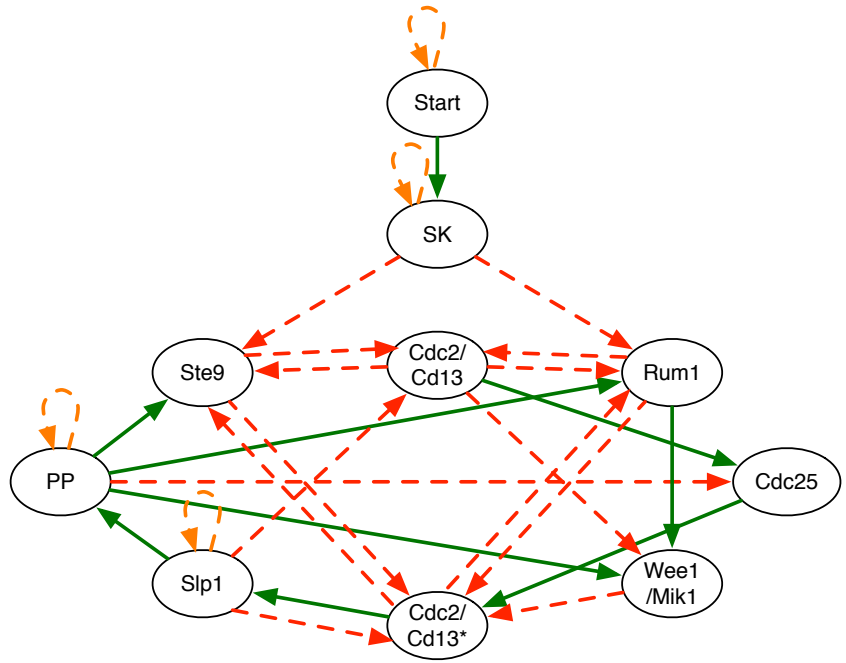


Neutral graph using 1000 functional networks

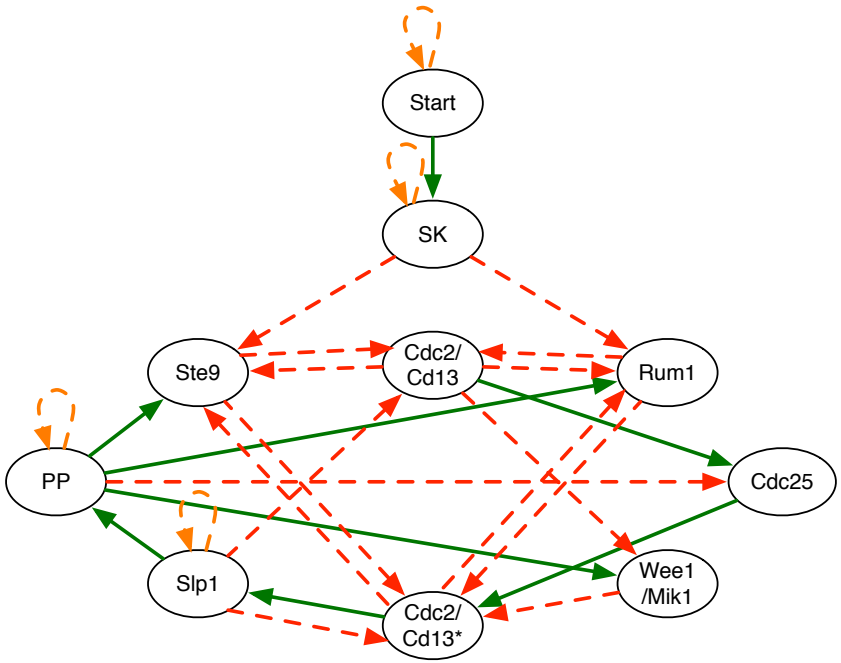


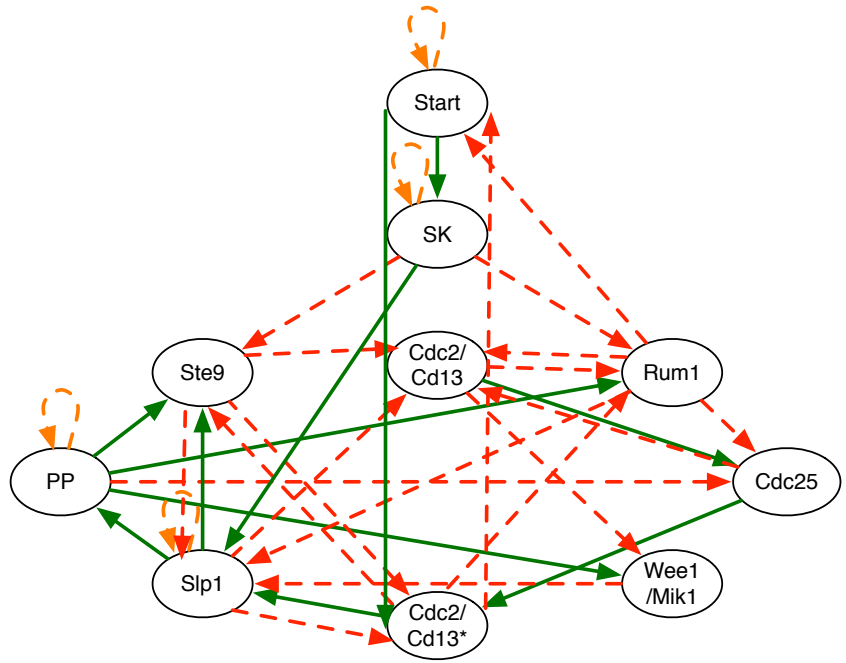
Wildtype network and a functional network in the connected component



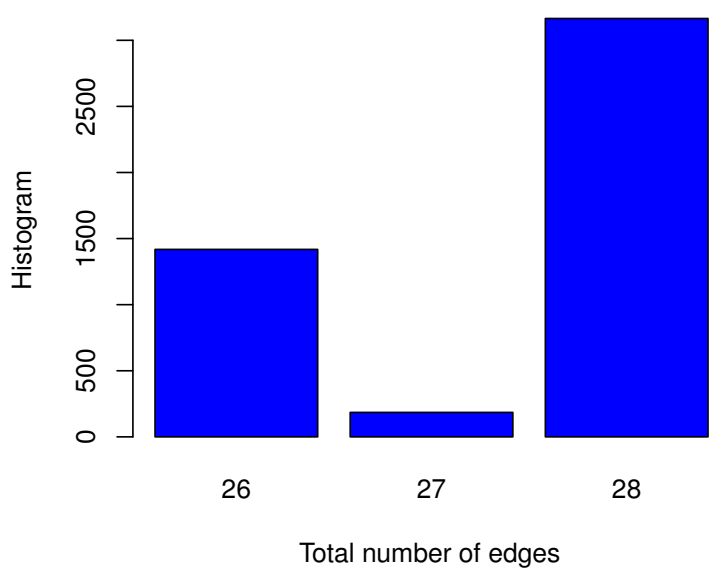


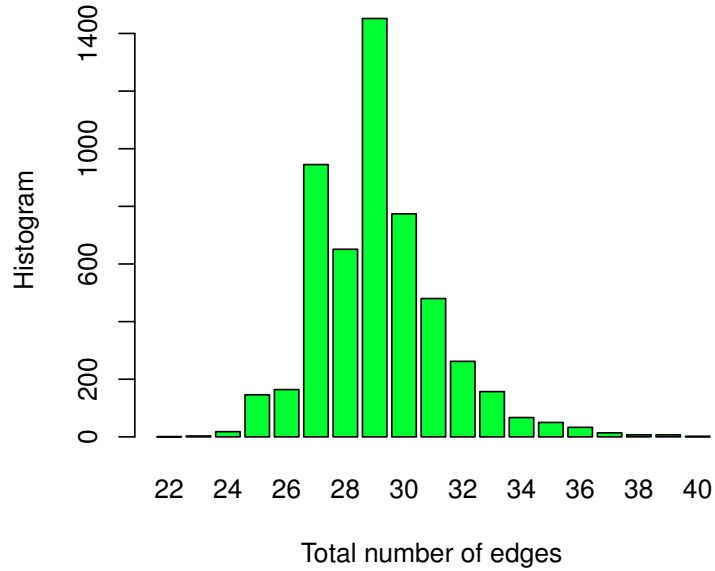
Wildtype network and a functional network not in the connected component



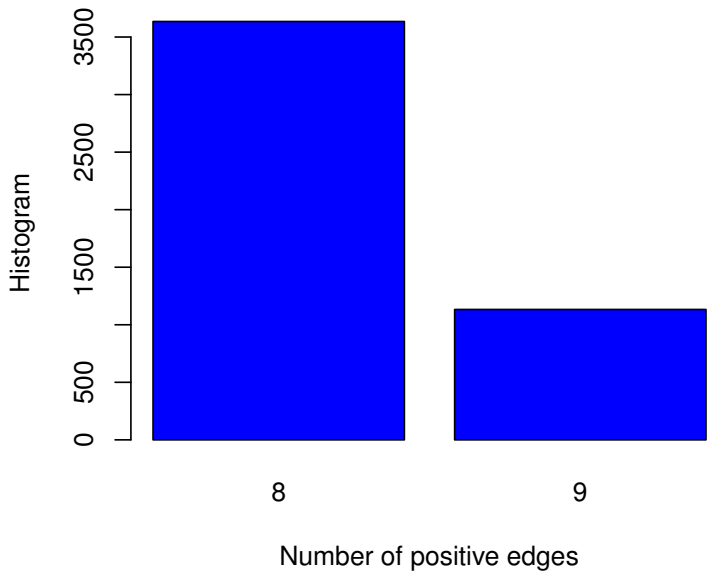


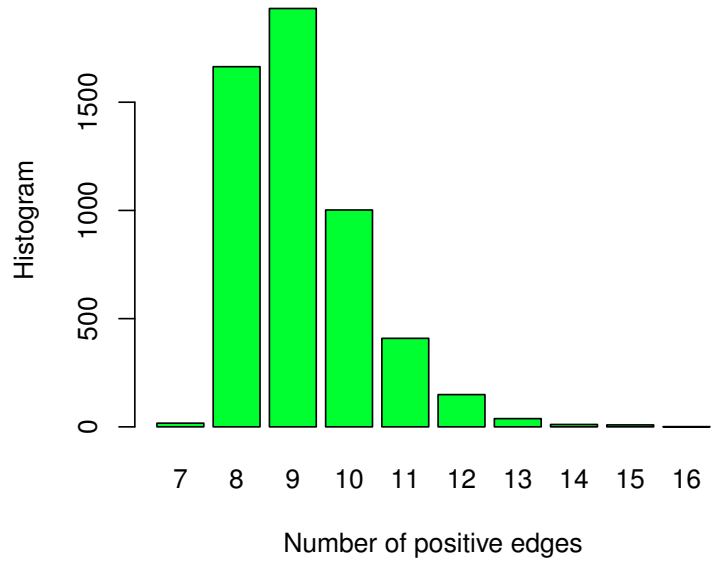
Connected component vs. not in the connected component (1)



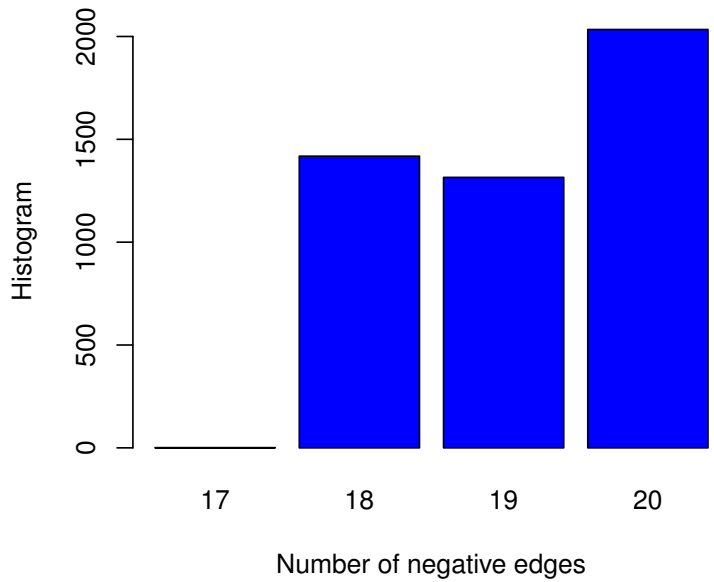


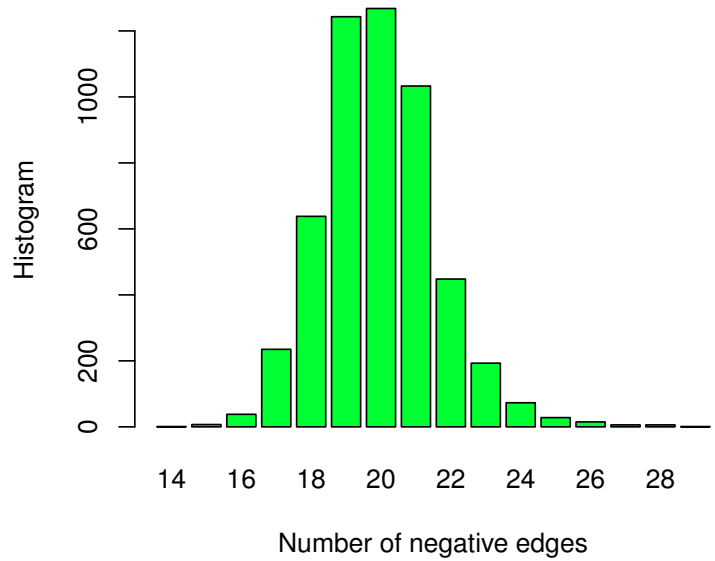
Connected component vs. not in the connected component (2)





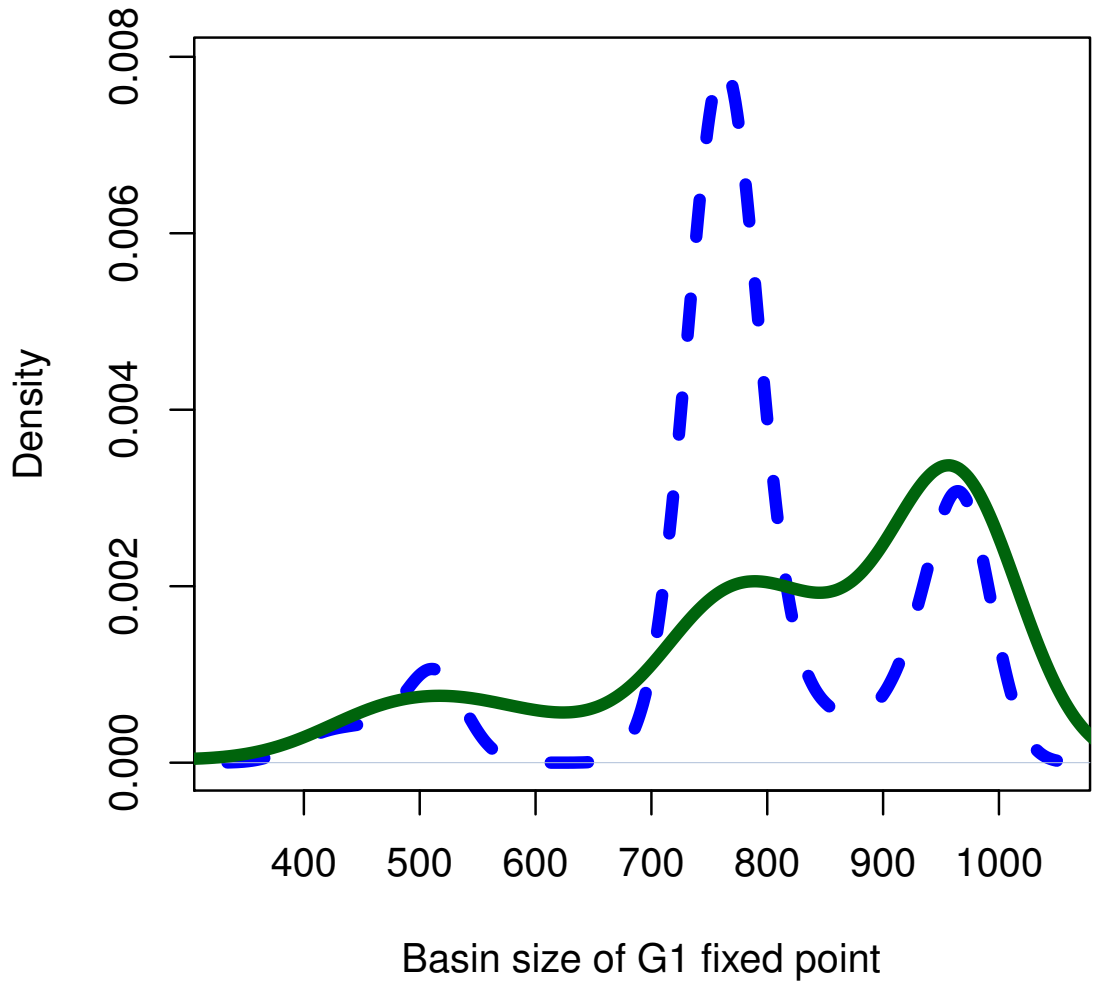
Connected component vs. not in the connected component (3)



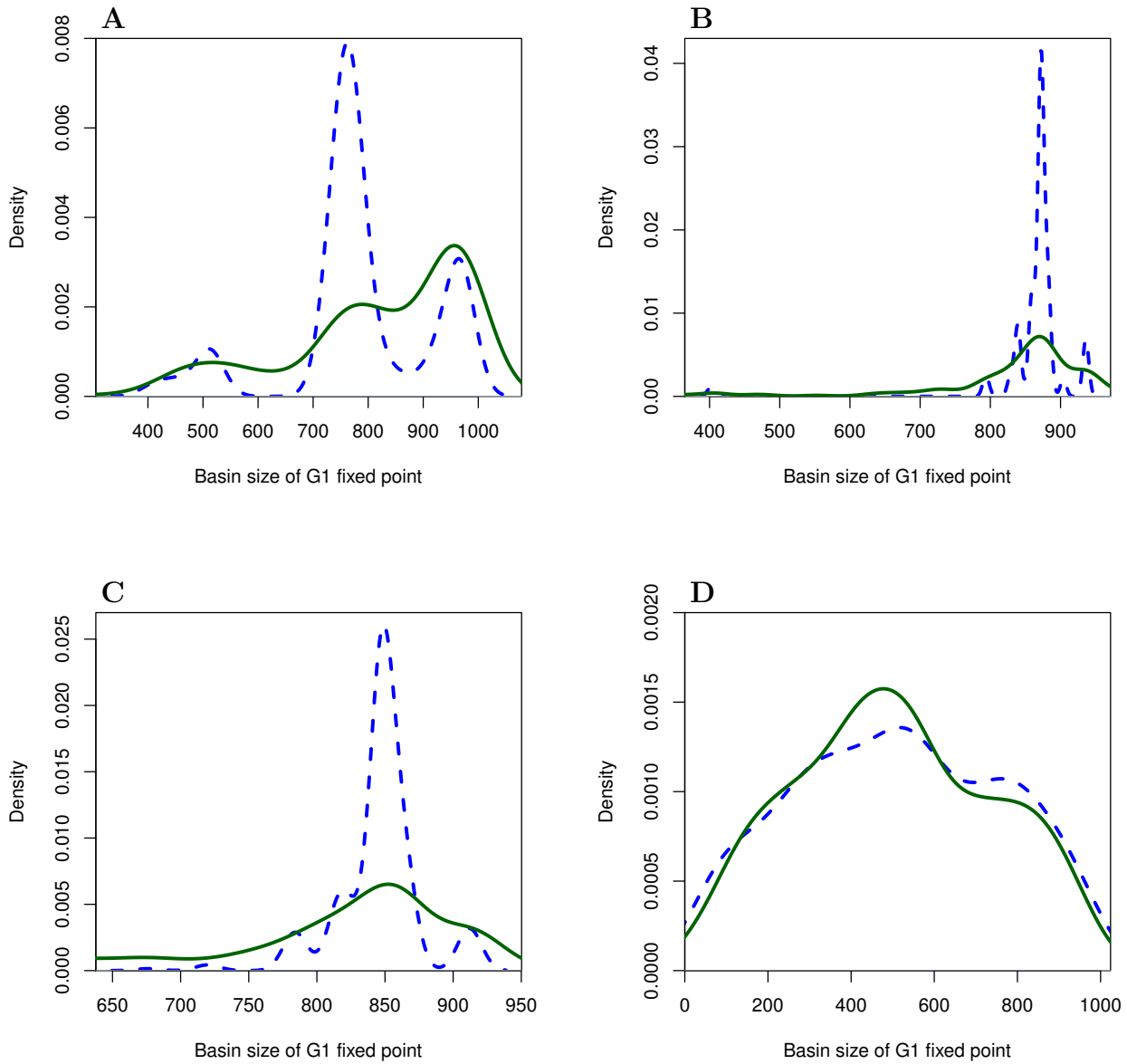


Density of the basin of attraction for the G_1 fixed point

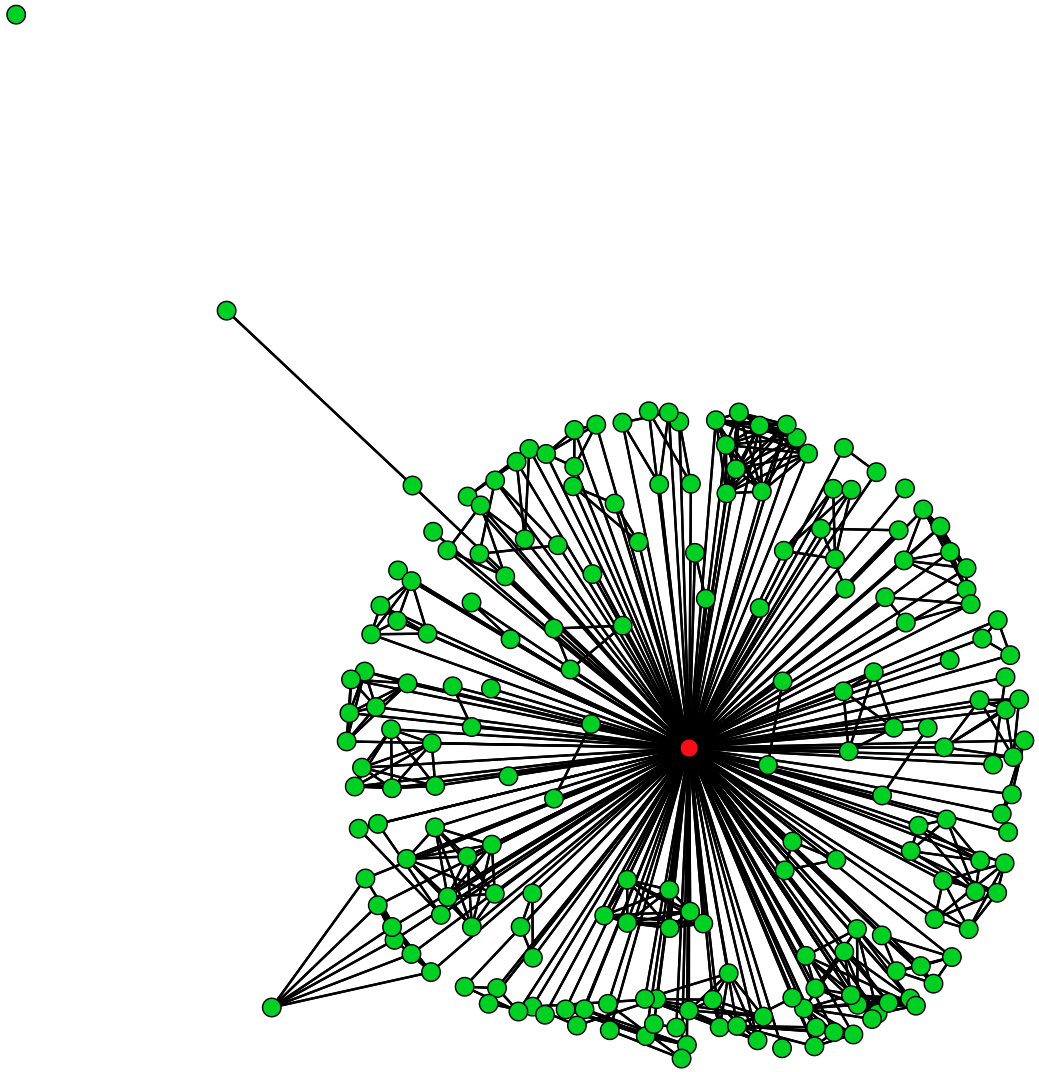
Functional networks in the wildtype connected component (blue/dashed line) and the rest of the networks (green/solid line)



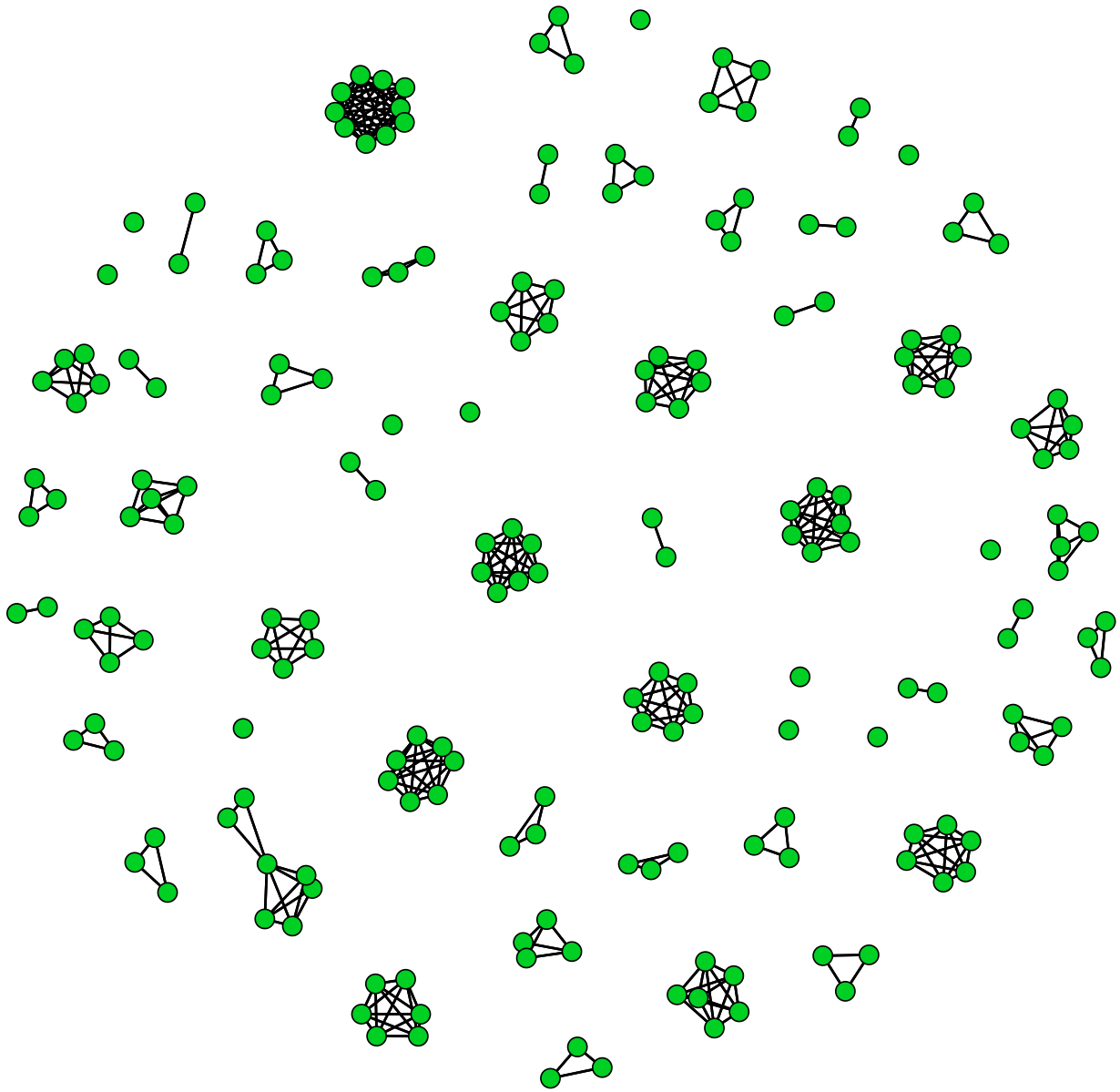
Other updating schemes?



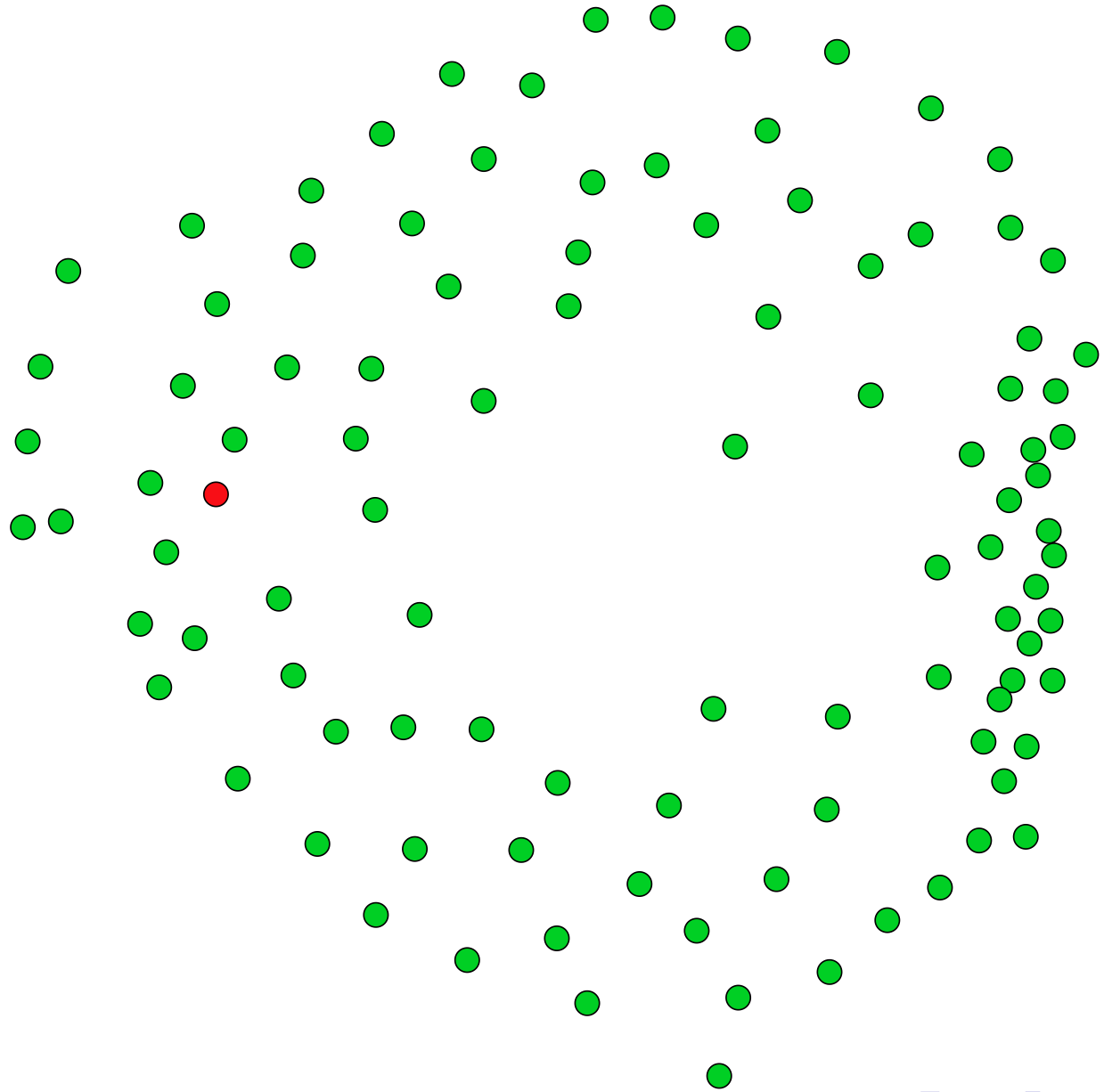
Neutral graph of the wildtype component



Neutral graph of the wildtype component without the wildtype network



Neutral graph generated using a genetic algorithm



Conclusion

- There are significant differences (topological and state space) between the functional networks in the connected component of the wildtype network and the rest of the network.

Conclusion

- There are significant differences (topological and state space) between the functional networks in the connected component of the wildtype network and the rest of the network.
- Functional networks in the wildtype connected component, can mutate up to no more than 3 times, then they reach a *point of no return* where the networks leave the connected component of the wildtype.

Conclusion

- There are significant differences (topological and state space) between the functional networks in the connected component of the wildtype network and the rest of the network.
- Functional networks in the wildtype connected component, can mutate up to no more than 3 times, then they reach a *point of no return* where the networks leave the connected component of the wildtype.
- The neutral space analysis may allow us to formulate new biological hypotheses studying the functional networks in the wildtype connected component, for example, analyzing which edges are in common, yielding a core structure that could explain the preservation of the functionality of the network.

Conclusion

- There are significant differences (topological and state space) between the functional networks in the connected component of the wildtype network and the rest of the network.
- Functional networks in the wildtype connected component, can mutate up to no more than 3 times, then they reach a *point of no return* where the networks leave the connected component of the wildtype.
- The neutral space analysis may allow us to formulate new biological hypotheses studying the functional networks in the wildtype connected component, for example, analyzing which edges are in common, yielding a core structure that could explain the preservation of the functionality of the network.
- Research problems: Sensitivity analysis, comparisons, neutral network disintegration, functionality.

References

- Ruz, G.A., Goles, E. Neutral graph of regulatory Boolean networks using evolutionary computation, The 2014 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2014), Honolulu, Hawaii, USA, May 21-24, 2014, pp. 1-8.
- Ruz, G.A., Timmermann, T., Barrera, J., Goles, E. Neutral space analysis for a Boolean network model of the fission yeast cell cycle network, *Biological Research*, Vol. 47, 2014, 64.
- Ruz, G.A., Timmermann, T., Goles, E. Neutral space analysis of gene regulatory network models of salt stress response in Arabidopsis using evolutionary computation. The 2016 IEEE Congress on Evolutionary Computation (IEEE CEC 2016), Vancouver, Canada, July 24-29, 2016, Accepted.

Thank you