Tracking the Evolution of Chemical Computing Networks

Thorsten Lenser, Naoki Matsumaru, Thomas Hinze, and Peter Dittrich

Bio Systems Analysis, Friedrich-Schiller-University Jena and Jena Centre for Bioinformatics
Institute of Computer Science, Ernst-Abbe-Platz 1-4, D-07743 Jena, Germany
{thlenser, naoki, hinze, dittrich}@minet.uni-jena.de

Abstract

How do chemical reaction networks that process information evolve? This is not only a fundamental question in the study of the origin of life, but also in diverse fields like molecular computing, synthetic biology, and systems biology. Here, we study the evolution of chemical flip-flops by means of chemical organisation theory. Additionally, we compare evolved circuits with manually constructed ones. We found that evolution selects for an organisational structure that is related to function. That is, the resulting computation can be explained as a transition between organisations. Furthermore, an evolutionary process can be tracked as a change of the organisational structure, which provides a fundamentally different view than looking at the structural changes of the reaction networks. In our experiments, 90% of evolutionary improvement coincide with a change in the organisational structure. We conclude that our approach provides a novel and useful perspective to study evolution of chemical information processing systems.

Introduction

In every living entity, cellular functions emerge from the astonishing interplay of connected reaction processes. Three essential types of biochemical networks can be distinguished: metabolic, cell signalling, and gene regulatory networks (Alberts et al., 2003). While metabolism consists of coupled enzymatically catalysed reactions supplying energy, cell signalling, and gene regulation perform information processing of external and internal signals (Cooper et al., 2001). Taking this information processing as a metaphor, biochemical reaction networks (or rather mathematical models of these) can be designed to perform specific computational tasks.

While a bottom-up approach has been pursued (Guido et al., 2006), top-down approaches, specifically evolutionary algorithms, have gained growing interests recently in order to design or program reaction systems. Efforts have been undertaken to evolve simple computational units (Deckard and Sauro, 2004), small biological networks (Koza et al., 2001; François and Hakim, 2004; Soyer et al., 2006), genetic regulatory networks (Dwight Kuo et al., 2006) or components thereof (Paladugu et al., 2006). Most of this work, however, has been focused on the final product, that is, the networks evolved to reproduce a certain specified behaviour. Here, we rather concentrate on the process of evolution. For that purpose, new methods are required that can deal with constructive systems (Fontana and Buss, 1994), that is, systems where new components (molecular species) or new interactions between existing components appear so that the network topology changes dynamically. Matsumaru et al. (2006) used chemical organisation theory (Dittrich and Speroni di Fenizio, 2007) in order to study the evolutionary dynamics of (artificial) chemical systems. In this paper, we analyse the trajectory of evolving chemical reaction networks that compute. That is, in particular, networks that function as flip-flops.

In previous work, the authors have developed a software designed to evolve biological networks (called the SBMLevolver) and measured the performance impact of certain design decisions for this algorithm (Lenser et al., 2007). That software package is adopted to evolve network models for this study. In the following section, the theory of chemical organisation is briefly reviewed. Then, the experimental setting to evolve a reaction network capable of flip-flop operation is presented. As results, three aspects of the evolutionary process are given in the Results section. In addition to the traditional aspect of the dynamical behaviour of the evolution, we analyse the dynamical change in terms of the chemical organisation within the reaction networks. We also show a reaction network evolved for the flip-flop function.

Reaction Networks and Chemical Organisations

Here, we utilized the notation of a chemical organisation developed by Dittrich and Speroni di Fenizio (2007) to analyse reaction networks. Following Fontana and Buss (1994), an organisation is defined as a set of molecular species that is closed and self-maintaining. The hierarchy of all organisations of a reaction network represents its organisational structure, which can be used to describe the dynamical (qualitative) behaviour of a reaction system as a movement between organisations (Speroni di Fenizio et al., 2001). Choosing a proper coding scheme, the organisational structure can be interpreted as a repertoire of behaviour patterns of the reaction system. For example, Dittrich and Speroni di Fenizio (2007) have shown that only species that form an organisation can makeup a stationary state.

The basic concepts needed here are now described more formally: A **reaction network** $\langle \mathcal{M}, \mathcal{R} \rangle$ consists of a set of (molecular) species $\mathcal{M}$ and a set of reaction rules $\mathcal{R}$. A **reaction rule** $\rho \in \mathcal{R}$ can be written according to the chemical notation:

$$l_{a_1,\rho}\, a_1 + l_{a_2,\rho}\, a_2 + \cdots \to r_{a_1,\rho}\, a_1 + r_{a_2,\rho}\, a_2 + \ldots \quad (1)$$

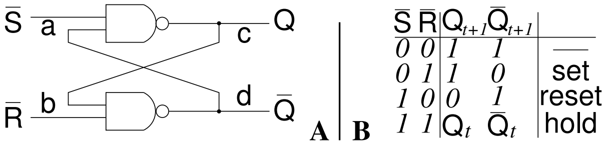
The **stoichiometric coefficients** $l_{a,\rho}$ and $r_{a,\rho}$ describe the amount of molecular species $a \in \mathcal{M}$ in reaction $\rho \in \mathcal{R}$ on the lefthand and righthand side, respectively. Together, the stoichiometric coefficients define the **stoichiometric matrix**

$$\mathbf{S} = (s_{a,\rho}) = (r_{a,\rho} - l_{a,\rho}). \quad (2)$$

An entry $s_{a,\rho}$ of the stoichiometric matrix denotes the net amount of molecules of type a produced in reaction ρ. We also define mappings $\mathrm{LHS}(\rho) \equiv \{a \in \mathcal{M} : l_{a,\rho} > 0\}$ and $\mathrm{RHS}(\rho) \equiv \{a \in \mathcal{M} : r_{a,\rho} > 0\}$, returning the species with a positive coefficient on the lefthand and righthand side, respectively. Reaction ρ can take place in $A \subseteq \mathcal{M}$ only when $\mathrm{LHS}(\rho) \subseteq A$.

Given a reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ with $m = |\mathcal{M}|$ species and $r = |\mathcal{R}|$ reactions, the organisational structure is derived with respect to the following two criteria: closure and self-maintenance. A set of species $A \subseteq \mathcal{M}$ is **closed**, if for all reactions ρ with $\mathrm{LHS}(\rho) \subseteq A$, the products are also contained in A, that is, $\mathrm{RHS}(\rho) \subseteq A$. This closure property ensures that there exists no reaction in A producing new species not yet present in the organisation using only species of that organisation. The other property is a theoretical capability of an organisation to maintain all of its members. Since the maintenance possibly involves complex reaction pathways, the stoichiometry of the whole reaction network must be considered in general. A set of molecules $C \subseteq \mathcal{M}$ is **self-maintaining**, if there exists a flux vector $\mathbf{v} \in \mathbb{R}^r$ such that the following three conditions apply: (1) for all reactions ρ that can take place in C (i.e., $\mathrm{LHS}(\rho) \subseteq C$) the flux $v_\rho > 0$; (2) for all remaining reactions ρ (i.e., $\mathrm{LHS}(\rho) \nsubseteq C$), the flux $v_\rho = 0$; and (3) for all molecules $a \in C$, the production rate $(\mathbf{Sv})_a \geq 0$. v_ρ denotes the element of $\mathbf{v}$ describing the flux (i.e. rate) of reaction ρ. $(\mathbf{Sv})_a$ is the production rate of molecule a given flux vector $\mathbf{v}$.

We visualize the set of all organisations with a Hasse diagram, in which organisations are arranged vertically according to their size in terms of the number of their members (cf. Figure 6). Two organisations are connected by a line if the upper organisation contains all species of the lower organisation and there is no other organisation between them. The Hasse diagram represents the hierarchical **organisational structure** of the reaction network under study.



S̄	R̄	Q_{t+1}	$\bar{Q}_{t+1}$	
0	0	1	1	—
0	1	1	0	set
1	0	0	1	reset
1	1	Q_t	$\bar{Q}_t$	hold

Figure 1: Circuit diagram and operation mode of flip-flop.

Method

We employ an evolutionary algorithm that instantiates a natural selection process on chemical reaction networks (Fernando and Rowe, 2007). The algorithm can mutate the reaction rules $\mathcal{R}$ of a reaction network with a fixed predefined set of molecular species $\mathcal{M} = \{a^0, a^1, b^0, b^1, c^0, c^1, d^0, d^1\}$. As mutational operators, the algorithm can add or delete a reaction, or replace a reaction with a different one, keeping as many of the previous participants as possible. To keep things simple, we employ a (1+1)-EA. That is, one parent generates one offspring, while the better of the two survives.

To enable neutral mutations and thus search space exploration, the offspring is kept if both have the same fitness. No parameter fitting is done, so that a change in parameters can only be realised through a replacement of a reaction with the same reaction, which has a different (randomly chosen) reaction constant. Only mass-action kinetics of first and second order are used in the evolution.

When speaking of a flip-flop logic gate in this work, we specifically mean an RS (Reset and Set) flip-flop, with a behaviour according to the truth table in Figure 1. To represent the four binary variables `a`, `b`, `c` and `d` making up this flip-flop in a chemical format, we employ two opposing species x^0 and x^1 for each binary variable `x`, where the presence of x^0 denotes the value `x = 0`, and x^1 denotes `x = 1` (cf. Matsumaru et al., 2007). To help maintain a valid state inside the system, we fix four destructive reactions $x^0 + x^1 \to \emptyset$ for all four species pairs $x^i = a^i, b^i, c^i, d^i$. These reactions cannot be changed or deleted by the evolutionary algorithm.

The ideal flip-flop that is the target of the artificial evolution works in the following way: The set operation $(\bar{S}, \bar{R}) = (0, 1)$ changes the state Q to 1, while the reset $(\bar{S}, \bar{Q}) = (1, 0)$ changes Q to 0. To hold the previous state, both inputs are set to 1. The forbidden input $(\bar{S}, \bar{Q}) = (0, 0)$ is not considered in the fitness function. In chemical form, the input $(\bar{S}, \bar{R}) = (0, 1)$ is represented by defining an inflow for a^0 and b^1, that is, $\{\emptyset \to a^0, \emptyset \to b^1\} \subseteq \mathcal{R}$; and the other two cases are treated similarly. The initial concentrations of c^i and d^i are set according to the previous state Q_t. Taking this together, we get six different test cases, coming from three different operations with two initial conditions each.

For each case, we specify either the presence or the absence of each species as desired, measured in steady state after simulating the reaction system for 1000 seconds. Numerical integration is done using the SBML ODE Solver Library (Machne et al., 2006). The classification as present or absent is decided by a concentration threshold of 10^{-9} (arbitrary units). For example, in the reset case, the following steady state concentrations are considered as correct: $a^1 = 1, a^0 = 0, b^1 = 0, b^0 = 1, c^1 = 0, c^0 = 1, d^1 = 1, d^0 = 0$. The fitness value is then calculated by counting the number of wrong presence / absence measurements, with 0 being the best possible fitness value. Once a fitness of 0 is reached, the evolution stops.

Results

To analyse the evolution of reaction networks acting as flip-flops, we performed 30 independent runs in order to evaluate properties of a "typical" run. Additionally, we also looked at one run in more detail.

Since the distinction between the three different input settings is realised by enabling or disabling inflow reactions for a^1, a^0, b^1 and b^0, we need to compute three lattices of organisations for each analysed candidate solution, one for each input setting.

Statistical analysis of many runs

The average fitness development (Figure 2) shows a stronger gain in fitness at the beginning, while the convergence towards zero is slower later on. Eventually, all runs reached a fitness of zero, i.e. the networks behaved as specified in the fitness function. Since a run stops exactly when the fitness of the current individual is 0, the number of generations usually differ between runs. In order to be able to average over these runs, we had to resample the data on fitness and number of organisations, such that a common number of measurements for each run is achieved. To this end, we constructed a timescale of "normalised evolutionary progress", defined by its endpoints 0.0 at the beginning of the evolution and 1.0 at the end when the final solution is found. The MATLAB function `resample`, which applies an anti-aliasing lowpass FIR filter during the resampling process, was used to create new data points at 1001 equally space points between 0.0 and 1.0.

Looking at the number of organisations for the three different input cases, we can see from Figure 3 that starting from around four to five organisations on average, the numbers diverge between the set/reset operations and the hold operation. While the number of organisations for the set/reset organisation converges between two and three, the hold operation yields around seven organisations on average.

By comparing the organisational structures between successive candidate solutions, we calculated that 90% of all fitness improvements are accompanied by a change in the organisational structure for at least one input case. In contrast, only 18% of organisational changes also come with a fitness improvement. When looking at the lineage of networks that led to the final solution, disregarding unsuccessful candidates, we find that 35% of all mutations changed the organisational structure for at least one input.

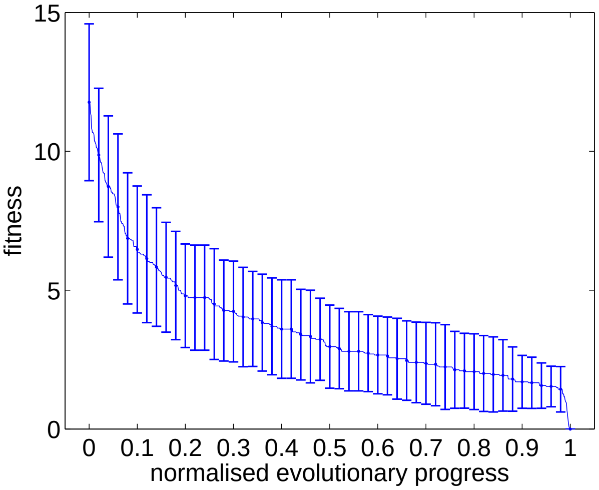


Figure 2: Average fitness value from beginning to end of evolutionary runs, from 30 independent repetitions. The x-axis denotes the normalised evolutionary progress from the random initial solution ($x = 0$) to finding a solution with fitness 0 ($x = 1$). For this, the different runs were resampled to 1000 samples, as described in the text. Errorbars indicate standard deviation.

Detailed analysis of one run

For an in-depth analysis, we pick the first evolutionary run that we performed for this problem. We will describe how the fitness improvements correlate with changes in the organisational structure, and give details on one specific mutational event and its consequences for the organisational structure of the network.

Comparing the average fitness development shown in Figure 2 with the single run analysed here (Figure 4 upper part), we can conclude that the fitness of the individual run progressed in a fairly standard way. This is especially true given that the behaviour of the 30 runs is quite diverse, as indicated by the large standard deviations. Also the length of this run (162 generations) is in the usual region, with an average run taking 221 generations with a standard deviation of 119. Also the number of organisations (Figure 4 lower part) is in agreement with the average number (Figure 3), even if the number of organisations for the set/reset operations are at the outer limits of the typical range (five and one, respectively).

Looking at the fitness increases in the course of the evolution (Figure 4 upper part) and the organisational structures of all networks that appear during the run (not shown), it can be observed that all but one of the eight fitness jumps are accompanied by changes in the organisational structure for at least one of the three input cases. However, taking the number of organisations at any point in the evolution into

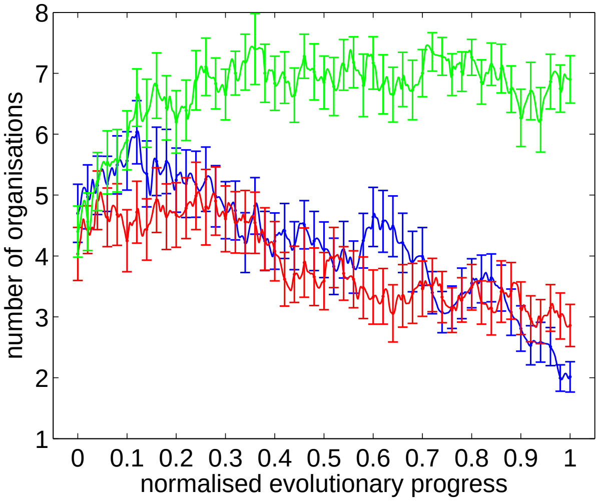


Figure 3: Average number of organisations from 30 independent runs of the evolution. Colors denote the (a_0, b_1) input (blue), the (a_1, b_0) input (red), and the (a_1, b_1) input (green). Errorbars indicate the standard error. Unit of x-axis as in figure 2.

account, one can also see that not every change in organisational structure leads to a fitness change, in fact, most do not.

We now relate the fitness change in one successful mutation to the change in organisational structure incurred by that mutation. As an example, we pick the fitness jump from generation 112 to 113, which improved the fitness from seven to four wrong presence/absence values. Looking at the reaction networks before and after the mutation (Figure 5), we see that the mutation added one reaction, which converts b^0 into d^1.

This additional reaction does not change the organisational structure for input cases (a^0, b^1) and (a^1, b^1) (set and hold, not shown), but reduces the lattice of organisations for input case (a^1, b^0) (reset) from five organisations to two (Figure 6). Looking at the behaviour of both networks for all input cases and initial configurations (i.e. all six test-cases), one can observe that the change occurs only in input case (a^1, b^0) with an initial configuration in which c^1 and d^0 are present (data not shown). For this case, the steady-state before the mutation has a^1, b^0, and c^1 present, and d^0 is still present after 1000 seconds even though its concentration is still decreasing at that time. This yields four wrong presence/absence values, since c^0 and d^1 should be present and c^1 and d^0 should be absent, but the total opposite is the case. After the mutation, d^1 is present and c^1 and d^0 are absent, but also c^0 is absent, so there is one wrong value left.

On the organisational level, the mutation removes three organisations (Figure 6), among which is also the organisation (a^1, b^0, c^1) responsible for the wrong behaviour of the network. After the mutation, the dynamics take the steady-state into organisation (a^1, b^0, d^1), resulting in a better behaviour. However, it is interesting to note that both organisational structures also contain organisation (a^1, b^0, c^0, d^1), which is the “correct” one that is also used in the final solution. Even though this organisation is present, the dynamics of both reaction systems are such that the steady-state does not lie inside it. We had to wait for another 49 generations for this to happen.

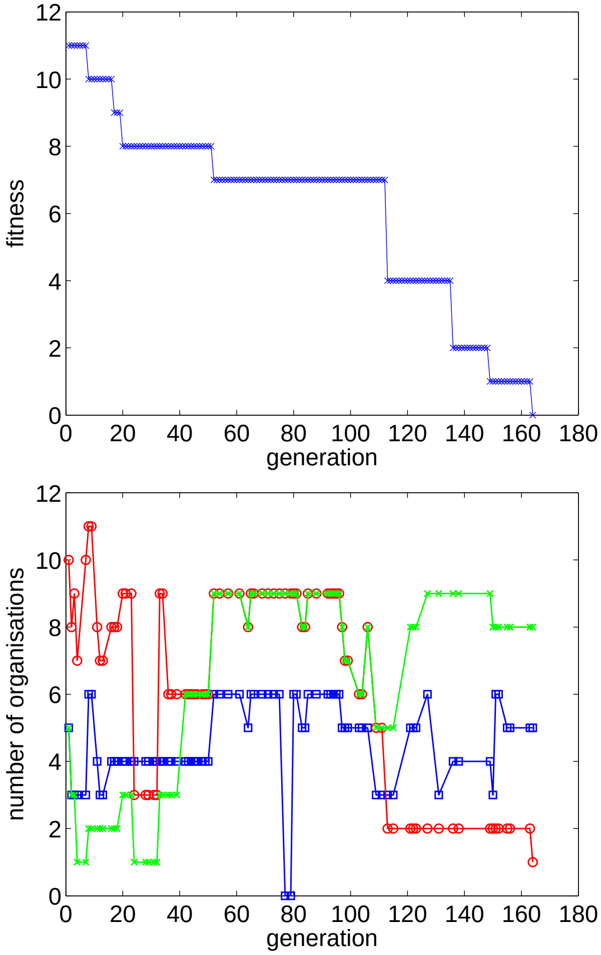


Figure 4: One exemplary run. Given are fitness (upper plot) and number of organisations for all three input cases (lower plot). In the lower plot, the three input cases are shown in blue ((a^0, b^1) input), red ((a^1, b^0) input), and green ((a^1, b^1)). Each mark (square, cross or circle) denotes a new network structure in the evolutionary trajectory.

An evolved chemical flip-flop

An outcome of the evolutionary process described above is analysed. The reaction network considered here has a fitness value of 0, i.e. solves the given task. The network structure is shown in Figure 7.

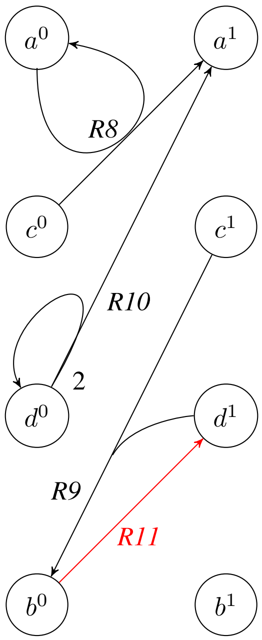


Figure 5: The reaction network of the candidate solution analysed in the text, after the mutation adding reaction *R11*. The added reaction is shown in red.

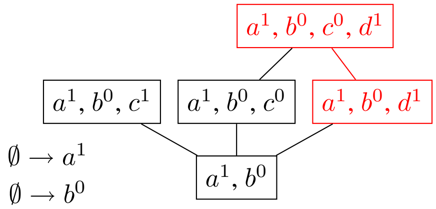


Figure 6: Organisational structure of the networks from Figure 5 for input (a^1, b^0), before (whole structure) and after addition of reaction *R11* (only red part).

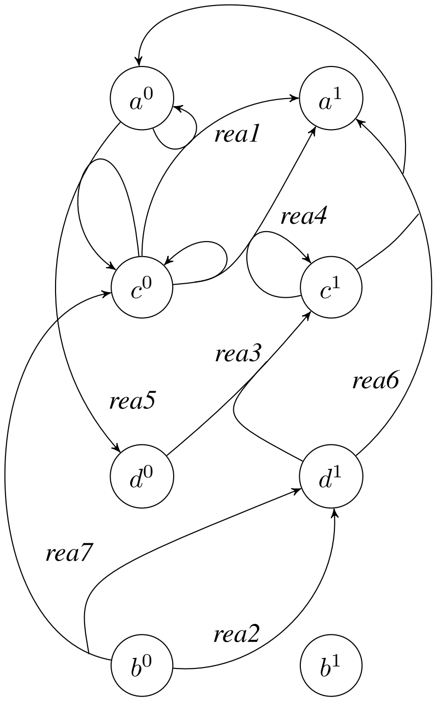


Figure 7: Chemical reaction network implementing flip-flop circuits, designed through an evolutionary process. Cooperative decay reactions ($a^1 + a^0 \to \emptyset$, $b^1 + b^0 \to \emptyset$, $c^1 + c^0 \to \emptyset$, $d^1 + d^0 \to \emptyset$) are omitted.

There are seven reactions, labeled as *rea1* to *rea7* in the figure, in addition to four reactions of cooperative decay (not shown in the figure), $a^1 + a^0 \to \emptyset$, $b^1 + b^0 \to \emptyset$, $c^1 + c^0 \to \emptyset$, $d^1 + d^0 \to \emptyset$. This base reaction network is extended to include inflow reactions, representing the inputs to the flip-flop circuit, depending on the operations. Organisational structures of the reaction system for each operational mode are shown in Figure 8.

Analysing the organisational structure of the reaction network, it becomes evident that the reaction system based on this reaction network is surely usable for the flip-flop computation. Including the two inflows $\emptyset \to a^1$ and $\emptyset \to b^0$ in the reaction network, as shown in Figure 8 A, only one set of species $\{a^1, b^0, c^0, d^1\}$ satisfies the conditions to be the organisation. It implies that only this species combination can be found in the dynamical reaction system in equilibrium states. Therefore, the reset operation can be realized in the evolved reaction system. The network with the inflows of $\emptyset \to a^0$ and $\emptyset \to b^1$ contains five organisations as shown in Figure 8 B, and one of those $\{a^0, b^1, c^1, d^0\}$ corresponds to the set operation.

Changing inflow reactions to $\emptyset \to a^1$ and $\emptyset \to b^1$ achieves the hold operation. In terms of the organisations, as shown in Figure 8 C, the two organisations *orgHR*=

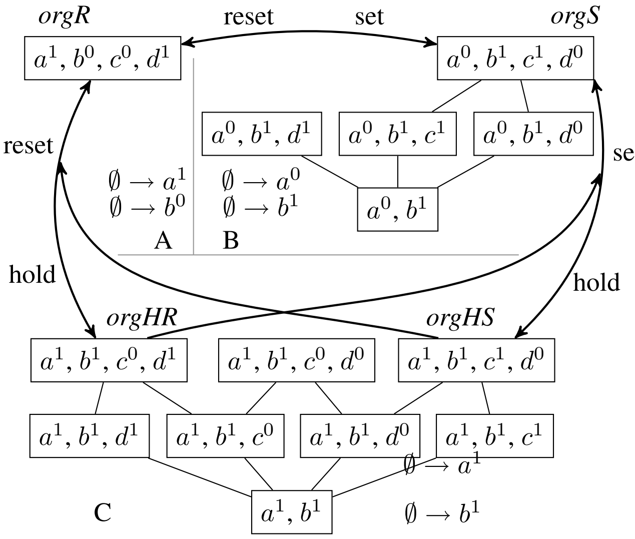


Figure 8: Organisational structure in the reaction network shown in Figure 7.

$\{a^1, b^1, c^0, d^1\}$ and *orgHS*= $\{a^1, b^1, c^1, d^0\}$ in the reaction network with those inflows reflect the bistability of the flip-flop circuit. Depending on the state at the previous time step, the hold operation results in a different state, namely the previous one. When the reaction system has been in the state after the set operation, (*i.e.*, *orgS*), the hold operation brings the system to the state of *orgHS*, keeping the output species unchanged as c^1 and d^0. Holding the information that the system has been reset can be achieved by moving the system state from *orgR* to *orgHR*.

The last operation of setting both inputs to be zero ($\mathtt{a} = \mathtt{b} = 0$) is forbidden for the flip-flop circuit. If adding two inflows of $\emptyset \to a^0$ and $\emptyset \to b^0$ to the base reaction network, one set of species becomes the organisation: $\{a^1, a^0, b^0, c^1, c^0, d^1, d^0\}$. Only b^1 is not involved to form the organisational structure.

If no inflow reaction is present, there are 42 organisations in the base reaction network. The smallest organisation is the empty set $\emptyset$. The sets containing four species forms the largest organisations, and there are four organisations of that size. The organisations with the size of four in Figure 8 are also found to be the organisation without inflows, except the organisation labeled as *orgR*. In fact, all organisations in Figure 8 except *orgR* are also organisations without inflows.

Dynamical Behaviour

To validate the organisational analysis of the reaction network, a dynamical reaction system is constructed and simulated with *Copasi* (Hoops et al., 2006), a biochemical reaction system simulator. Agreeing to the fitness calculation of the evolutionary design process, mass action kinetics is assumed for every reaction, if applicable. The ordinary differential equations (ODEs) for the input species read:

$$\begin{aligned} \dot{[a^1]} &= k_1[a^0][c^0] + k_4[c^1][c^0] + k_6[c^1][d^1] \\ &\quad -d_a[a^1][a^0] + I_{a^1}(1-[a^1]) \qquad (3) \\ \dot{[a^0]} &= -k_5[a^0][c^0] + k_6[c^1][d^1] \\ &\quad -d_a[a^1][a^0] + I_{a^0}(1-[a^0]) \qquad (4) \\ \dot{[b^1]} &= -d_b[b^1][b^0] + I_{b^1}(1-[b^1]) \qquad (5) \\ \dot{[b^0]} &= -k_2[b^0] - k_7[b^0] \\ &\quad -d_b[b^1][b^0] + I_{b^0}(1-[b^0]) \qquad (6) \end{aligned}$$

where a kinetic parameter for a reaction rea_id is denoted as k_{rea_id}. Kinetic parameters for the cooperative decay reactions are represented by d, and the subscript specifies the pair. For example, the decay rate of the cooperative decay reaction $a^0 + a^1 \to \emptyset$ is denoted as d_a.

Inflow reactions representing the operation of reset, set, and hold are controlled by the four parameters: I_{a^1}, I_{a^0}, I_{b^1}, and I_{b^0}. These parameters are binary variables, accepting only 0 or 1. For example, when the chemical flip-flop is set, I_{a^0} and I_{b^1} are set to one and the other pair of parameters $I_{a^1} = I_{b^0} = 0$ is set to zero. Inflows are assumed to be constant fluxes. Furthermore, the inflows are linked to normal decay reactions such as $a^1 \to \emptyset$ in order to avoid endless increase of the input species concentration. The resulting term of the ODE is $I_{a^1}(1-[a^1])$, for example.

The ODEs for the output species read:

$$\begin{aligned} \dot{[c^1]} &= k_3[d^1][d^0] - k_6[d^1][c^1] \\ &\quad -d_c[c^1][c^0] - I_{b^0}[c^1] \qquad (7) \\ \dot{[c^0]} &= -k_1[a^0][c^0] + k_7[b^0] \\ &\quad -d_c[c^1][c^0] - I_{b^0}[c^0] \qquad (8) \\ \dot{[d^1]} &= k_2[b^0] - k_3[d^1][d^0] - k_6[d^1][c^1] + k_7[b^0] \\ &\quad -d_d[d^1][d^0] - I_{b^0}[d^1] \qquad (9) \\ \dot{[d^0]} &= -k_3[d^1][d^0] + k_5[a^0][c^0] \\ &\quad -d_d[d^1][d^0] - I_{b^0}[d^0] \qquad (10) \end{aligned}$$

Kinetic parameter values are also provided as the outcome of the evolutionary design, but we manually adjusted the values so that the operations can be continuously repeated. When the fitness of the reaction system was calculated during the evolution process, three of the operations were evaluated separately and the reaction system was reinitialized for each case. This re-initialization step between operations is prevented so that the end state of the previous operation becomes the initial state of the next operation. For that purpose, the outflows of the input species are added as described above in order to restrict the increase of the concentration. For the output species, the outflows are also added as shown above, activated only when the inflow of b^0 is present. This modification is also to restrict the increase of the concentrations of the output species, specially, when the system is reset.

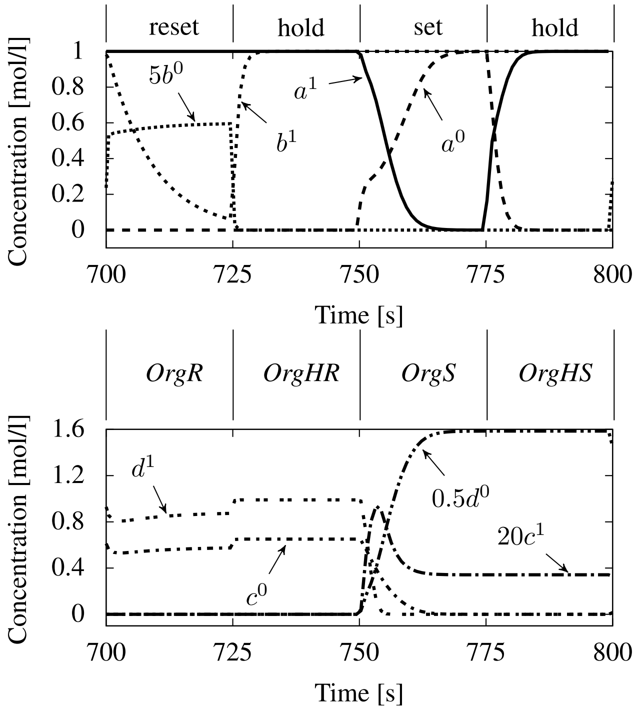


Figure 9: Dynamical simulation of chemical flip-flop designed by evolution. Parameters are set as follows: $d_a = d_b = 0.1$, $k_4 = 2.33941$, $k_6 = 2.83745$, $k_1 = 4.44231$, $k_5 = 3.62963$, $k_7 = 4.82838$, $k_3 = 1.0$, $k_2 = 0.1$, $d_c = 0.001$, $d_d = 1.0$. Additionally, for each operation of reset, set, and hold, inflow reactions are activated. For the set operation, the parameters are set such that $I_{a^0} = I_{b^1} = 0$ and $I_{a^1} = I_{b^0} = 1$ to activate inflows of a^1 and b^0 species and to deactivate the others. The reset operation is initiated by setting $I_{a^0} = I_{b^1} = 1$ and $I_{a^1} = I_{b^0} = 0$. The hold operation is achieved with the parameter settings of $I_{a^1} = I_{b^1} = 1$ and $I_{a^0} = I_{b^0} = 0$.

The last modification is the kinetic parameter of the reaction *rea1*, k_1, from 4.44231 to 0.5. The rational of this adjustment is: under the input condition "set", the system is observed to converge to the organisation of $\{a^0, b^1, d^1\}$, instead of *orgS*. This behaviour results from the fast extinction of species c^0 so that the generation of d^0 by *rea5* is insufficient. Slowing down the reaction speed of *rea1*, species c^0 stays in the system longer and produces d^0 enough to neutralize d^1.

Conclusion

We found that most fitness improvements come together with change in organisational structure (90%), showing that organisation analysis indeed yields insight into the evolutionary process. On the other hand, most organisational changes are fitness-neutral (82%), indicating that a lot of the information given in the lattice of organisations does not directly relate to the measured function of the networks. We have also seen mutations where the replacement of a reaction with the same type of reaction led to a fitness increase caused purely by the changing of a kinetic parameter, as well as changes of network structure not reflected in the organisations (but improving fitness). All this implies that while organisational analysis can give us many indications regarding the function of a reaction network, sometimes it does not tell the whole story of the network's dynamics.

We have also seen that the number of organisations for the set and reset states is substantially smaller than the number for the hold state, in analogy to the hand-constructed flip-flop by Matsumaru et al. (2007). In comparison to their solution, the evolved networks show a larger number of organisations for each input case. To realize the flip-flop behaviour in the reaction system, the minimum number of organisations in the reaction network is one for the set and reset operation and three for the hold operation. The hand-designed flip-flop implementation shown by Matsumaru et al. (2007) has two organisations for set and reset, respectively, and three for hold. In comparison, the evolved networks have more organisations, on average between two and three each for set and reset, and seven for hold. This implies that even though the function of the flip-flop networks is reflected in their organisational structure, this structure contains more information than only the operational modes specified in the fitness function.

As an interesting extension to this work, one could use organisational analysis to direct the evolution of reaction networks. By first designing the perfect organisational structure and then evolving networks with this structure, it would be possible to study whether these networks have the desired functionality. A key step in this direction is certainly the design of an appropriate fitness function based on a network's lattice of organisations.

In an additional investigation on top of the results shown here, one should look at the effect of different mutational operators on network structure, fitness and organisational structure. This will lead to helpful insights on how the mutations affect the lattice of organisations, and also on how specific organisational changes are related to changes in the fitness function.

In our opinion, the most important lesson to be learned from this work is that the evolutionary process investigated here produces reaction networks with an organisational structure that reflects their flip-flop functionality. Even though our choice of representation format of the binary information in chemical form may favour this, we believe that this phenomenon is mainly caused by the structure of the fitness function, i.e. by the task that is required of the networks. It will be very interesting in future to investigate this with other representation formats.

Acknowledgements

We acknowledge financial supports by the European Union, NEST-project ESIGNET no. 12789 and by the German Research Foundation (DFG) Grant DI 852/4.

References

Alberts, B., Johnson, A., and Lewis, J. (2003). *Essential Cell Biology*. Garland Publishing.

Cooper, B., Schonbrunner, N., and Krauss, G. (2001). *Biochemistry of signal transduction and regulation*. Wiley-VCH.

Deckard, A. and Sauro, H. (2004). Preliminary studies on the in silico evolution of biochemical networks. *ChemBioChem*, 5:1423–1431.

Dittrich, P. and Speroni di Fenizio, P. (2007). Chemical organisation theory. *Bull Math Biol*, 69(4):1199–1231.

Dwight Kuo, P., Banzhaf, W., and Leier, A. (2006). Network topology and the evolution of dynamics in an artificial genetic regulatory network model created by whole genome duplication and divergence. *BioSystems*, 85:177–200.

Fernando, C. and Rowe, J. (2007). Natural selection in chemical evolution. *Journal of Theoretical Biology*, 247(1):152–167.

Fontana, W. and Buss, L. W. (1994). 'The arrival of the fittest': Toward a theory of biological organization. *Bull Math Biol*, 56:1–64.

François, P. and Hakim, V. (2004). Design of genetic networks with specified functions by evolution in silico. *PNAS*, 101:580–585.

Guido, N. J., Wang, X., Adalsteinsson, D., McMillen, D., Hasty, J., Cantor, C. R., Elston, T. C., and Collins, J. J. (2006). A bottom-up approach to gene regulation. *Nature*, 439(7078):856–860.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006). COPASI - a COmplex PAthway SImulator. *Bioinformatics*, 22:3067–3074.

Koza, J., Mydlowec, W., Lanza, G., Yu, J., and Keane, M. (2001). Automatic synthesis of both the topology and sizing of metabolic pathways using genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 57–65. Morgan Kaufmann.

Lenser, T., Hinze, T., Ibrahim, B., and Dittrich, P. (2007). Towards evolutionary network reconstruction tools for systems biology. In E. Marchiori, J.H. Moore, J. R. E., editor, *Proceedings of the Fifth European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBIO)*, volume 4447 of *LNCS*.

Machne, R., Finney, A., Muller, S., Lu, J., Widder, S., and Flamm, C. (2006). The SBML ODE Solver Library: a native API for symbolic and fast numerical analysis of reaction networks. *Bioinformatics*, 22(11):1406–7.

Matsumaru, N., Centler, F., Speroni di Fenizio, P., and Dittrich, P. (2007). Chemical organization theory as a theoretical base for chemical computing. *International Journal of Unconventional Computing*, 3(4):285–309.

Matsumaru, N., Speroni di Fenizio, P., Centler, F., and Dittrich, P. (2006). On the evolution of chemical organizations. In Artmann, S. and Dittrich, P., editors, *Explorations in the complexity of possible life: abstracting and synthesizing the principles of living systems, Proceedings of the 7th German Workshop of Artificial Life*, pages 135–146. Aka, Berlin.

Paladugu, S., Chickarmane, V., Deckard, A., Frumkin, J., McCormack, M., and Sauro, H. (2006). In silico evolution of functional modules in biochemical networks. *IEE Proceedings-Systems Biology*, 153(4):223–235.

Soyer, O., Pfeiffer, T., and Bonhoeffer, S. (2006). Simulating the evolution of signal transduction pathways. *Journal of Theoretical Biology*, 241:223–232.

Speroni di Fenizio, P., Dittrich, P., and Banzhaf, W. (2001). Spontanous formation of cells in a universal artificial chemistry on a planar graph. In Kelemen, J. and Sosik, P., editors, *Advances in Artificial Life. Proc. 6th European Conference on Artificial Life (ECAL 2001), Prague, Czech Republic, September 10 - 14, 2001, LNCS 2159*, pages 206–215. Springer, Berlin.