Supplementary data:  
Towards dynamic genome scale models  

David Gilbert¹, Monika Heiner¹,², Yasoda Jayaweera¹, and Christian Rohr²

¹ Brunel University London, UK  
² Brandenburg Technical University, Cottbus, Germany  
{david.gilbert,yasoda.jayaweera}@brunel.ac.uk  
{monika.heiner,christian.rohr}@b-tu.de

Section A – Formal Petri Net Definitions

We recall basic terminology of Petri net theory; see [1] for further details.

Definition 1 (Petri net, syntax). A Petri net is a tuple \( N = (P,T,f,m_0) \), where

- \( P \) and \( T \) are finite, non-empty, and disjoint sets. \( P \) is the set of \textit{places}, and \( T \) is the set of \textit{transitions}.
- \( f : ((P \times T) \cup (T \times P)) \rightarrow \mathbb{N}_0 \) defines the set of directed \textit{arcs}, weighted by non-negative integer values.
- \( m_0 : P \rightarrow \mathbb{N}_0 \) gives the \textit{initial marking}.

We introduce the following notions and notations.

- \( m(p) \) yields the number of tokens on place \( p \) in the marking \( m \). A place \( p \) with \( m(p) = 0 \) is called \textit{unmarked} in \( m \), otherwise it is called \textit{marked}. A set of places is called unmarked if all its places are unmarked, otherwise marked.
- The preset of a node \( x \in P \cup T \) is defined as \( \bullet x := \{ y \in P \cup T | f(y,x) \neq 0 \} \), and its postset as \( x^* := \{ y \in P \cup T | f(x,y) \neq 0 \} \). We obtain four types of sets:
  - \( \bullet t \), the pre-places of a transition \( t \), consisting of the reaction’s substrates,
  - \( t^* \), the post-places of a transition \( t \), consisting of the reaction’s products,
  - \( \bullet p \), the pre-transitions of a place \( p \), consisting of all reactions producing this metabolite,
  - \( p^* \), the post-transitions of a place \( p \), consisting of all reactions consuming this metabolite.

We extend both notions to a set of nodes \( X \subseteq P \cup T \) and define:

- the set of all pre-nodes \( \bullet X := \bigcup_{x \in X} \bullet x \), and
- the set of all post-nodes \( X^* := \bigcup_{x \in X} x^* \).

A node \( x \in P \cup T \) is called \textit{boundary node} if \( \bullet x = \emptyset \lor x^* = \emptyset \); there are four types.

- A place \( p \) is called \textit{source place}, if \( \bullet p = \emptyset \).
- A place \( p \) is called \textit{sink place}, if \( p^* = \emptyset \).
- A transition \( t \) is called \textit{source transition}, if \( \bullet t = \emptyset \).
A transition $t$ is called **sink transition**, if $t^* = \emptyset$.

Boundary nodes model interconnections of an open system with its environment.

**Definition 2 (Petri net, semantics).** Let $\mathcal{N} = (P, T, f, m_0)$ be a Petri net.

- A transition $t$ is enabled in a marking $m$, written as $m \models t$, if
  \[ \forall p \in t : m(p) \geq f(p, t), \] else **disabled**.
- A transition $t$, which is enabled in $m$, may fire.
- When $t$ in $m$ fires, a new marking $m'$ is reached, written as $m \xrightarrow{t} m'$, with
  \[ \forall p \in P : m'(p) = m(p) - f(p, t) + f(t, p). \]
- The firing happens atomically.

In qualitative (time-free) Petri nets, the firing does not consume any time, while in quantitative (stochastic, continuous, hybrid) Petri nets, transitions are associated with generally state-dependent firing rates.

The net structure can be encoded by a matrix called incidence matrix in the Petri net community, and stoichiometric matrix in systems biology [2].

**Definition 3 (P/T-invariants).**

- The **incidence matrix** of $\mathcal{N}$ is a matrix $C : P \times T \to \mathbb{Z}$, indexed by $P$ and $T$, such that $C(p, t) = f(t, p) - f(p, t)$.
- A **place vector** (transition vector) is a vector $x : P \to \mathbb{Z}$, indexed by $P$,
  and $y : T \to \mathbb{Z}$, indexed by $T$.
- A place vector (transition vector) is called a **P-invariant** (T-invariant) if it is
  a nontrivial nonnegative integer solution of the linear equation system $x \cdot C = 0$
  ($C \cdot y = 0$).
- The set of nodes corresponding to an invariant’s nonzero entries are called the support
  of this invariant $x$, written as $\text{supp}(x)$.
- An invariant $x$ is called **minimal** if no invariant $z$ : $\text{supp}(z) \subset \text{supp}(x)$, and the
greatest common divisor of all nonzero entries of $x$ is 1.
- A net is covered by P-invariants, shortly CPI, (covered by T-invariants, shortly CTI) if every place (transition) belongs to a P-invariant (T-invariant).

The two transitions corresponding to a reversible reaction form a trivial T-invariant. Minimal non-trivial T-invariants correspond to elementary modes [2]; both notions coincide for networks without reversible reactions.

**State equation.** A transition vector $y$ can be read as counting vector (Parikh vector) of a firing word (transition sequence) $w \in T^*$, leading from $m$ to $m'$, with $m, m' \in \mathbb{N}_{\geq 0}^{|P|}$, written as $m \xrightarrow{w} m'$. Then,

\[ C' \cdot y \]

give the total change on a marking by firing the transition sequence $w$, and

\[ m' = m + C' \cdot y \]

gives the relation between the start marking $m$ and the target marking $m'$.

Finally, we recall the notion of **induced subnet**.
Definition 4 (Place-induced subnet). Let $\mathcal{N} = (P, T, f, m_0)$ be a Petri net and $S$ a set of places, $S \subseteq P$. The subnet $\mathcal{N}_S = (P_S, T_S, f_S, m_{0S})$ of $\mathcal{N}$ induced by $S$ is defined by

- $P_S = S$,
- $T_S = S \cup S^*$,
- $f_S = f \cap (P_S \cup T_S) \times (P_S \cup T_S)$,
- $m_{0S}$ is the projection of $m_0$ to $P_S$.

Definition 5 (Transition-induced subnet). Let $\mathcal{N} = (P, T, f, m_0)$ be a Petri net and $R$ a set of transitions, $R \subseteq T$. The subnet $\mathcal{N}_R = (P_R, T_R, f_R, m_{0R})$ of $\mathcal{N}$ induced by $R$ is defined by

- $T_R = R$,
- $P_R = R \cup R^*$,
- $f_R = f \cap (P_R \cup T_R) \times (P_R \cup T_R)$,
- $m_{0R}$ is the projection of $m_0$ to $P_R$.

P-invariants correspond to token-conserving net components, defined by the subnets induced by the P-invariants’ supports, and T-invariants correspond to state-reproducing net components, defined by the subnets induced by the T-invariants’ supports.
Section B – P-invariants of the running example

We give the P-Invariants in the original format as generated by Charlie [3].

minimal semipositive place invariants=

1 | 12.M_adp_c :1,
   | 15.M_amps_c :1,
   | 16.M_atp_c :1
2 | 49.M_nad_c :1,
   | 50.M_nadh_c :1
3 | 9.M_accoa_c :1,
   | 20.M_coa_c :1,
   | 70.M_succoa_c :1
4 | 51.M_nadp_c :1,
   | 52.M_nadph_c :1
5 | 63.M_q8_c :1,
   | 64.M_q8h2_c :1

How to read. There are five P-invariants, which happen not to overlap. The first P-invariant comprises three places, all have a weight of 1. The number preceding a place name gives the node number (required for internal purposes). Likewise for the other invariants.
Section C – Meta model

The meta model contains all reactions which can possibly be used in any configuration, and these can be turned on or off by setting the associated kinetic parameters to non-zero or zero. The parameter declarations can be arranged in named groups (e.g. growth, aerobic, typofix), with possible values arranged in named value sets (e.g. on, off). This permits model configurations in the style of:

```
modelname.andl growth=off,aerobic=on,typofix=on
```

For more details see the Marcie manual [4].

The meta model itself together with the scripts that we used can be downloaded from [http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Examples](http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Examples).
Section D – Property libraries

D.1 Reaction library

\[ P \geq 1 \left[ G \left( [\lt x\rt] = 0 \right) \right] \]
% 01_never_active

\[ P \geq 1 \left[ (\text{time} > 0 \& \text{time} \leq 250) \rightarrow G \left( [\lt x\rt] = 0 \right) \right] \]
% 01a_time_never_active_0to250

\[ P \geq 1 \left[ (\text{time} > 250 \& \text{time} \leq 500) \rightarrow G \left( [\lt x\rt] = 0 \right) \right] \]
% 01b_time_never_active_250to500

\[ P \geq 1 \left[ (\text{time} > 500 \& \text{time} \leq 750) \rightarrow G \left( [\lt x\rt] = 0 \right) \right] \]
% 01c_time_never_active_500to750

\[ P \geq 1 \left[ (\text{time} > 750 \& \text{time} \leq 1000) \rightarrow G \left( [\lt x\rt] = 0 \right) \right] \]
% 01d_time_never_active_750to1000

\[ P \geq 1 \left[ F \left( [\lt x\rt] > 0 \right) \right] \]
% 02_sometime_active

\[ P \geq 1 \left[ (\text{time} > 0 \& \text{time} \leq 250) \rightarrow F \left( [\lt x\rt] > 0 \right) \right] \]
% 02a_sometime_active

\[ P \geq 1 \left[ (\text{time} > 250 \& \text{time} \leq 500) \rightarrow F \left( [\lt x\rt] > 0 \right) \right] \]
% 02b_sometime_active

\[ P \geq 1 \left[ (\text{time} > 500 \& \text{time} \leq 750) \rightarrow F \left( [\lt x\rt] > 0 \right) \right] \]
% 02c_sometime_active

\[ P \geq 1 \left[ (\text{time} > 750 \& \text{time} \leq 1000) \rightarrow F \left( [\lt x\rt] > 0 \right) \right] \]
% 02d_sometime_active

\[ P \geq 1 \left[ G \left( d[\lt x\rt] = 0 \& [\lt x\rt] > 0 \right) \right] \]
% 03_always_steadystate_active_abovezero

\[ P \geq 1 \left[ (\text{time} > 900 \& \text{time} \leq 1000) \rightarrow G \left( d[\lt x\rt] = 0 \& [\lt x\rt] > 0 \right) \right] \]
% 03b_always_steadystate_active_abovezero900to1000

\[ P \geq 1 \left[ G \left( d[\lt x\rt] = 0 \right) \right] \]
% 04_always_steadystate_active_any_value

\[ P \geq 1 \left[ F \left( G \left( [\lt x\rt] = 0 \& d[\lt x\rt] = 0 \right) \right) \& F \left( d[\lt x\rt] \neq 0 \right) \right] \]
% 04a_always_steadystate_active_abovezero

\[ P \geq 1 \left[ F \left( G \left( [\lt x\rt] > 0 \& d[\lt x\rt] = 0 \right) \right) \& F \left( d[\lt x\rt] \neq 0 \right) \right] \]
% 04b_always_steadystate_active_abovezero

\[ P \geq 1 \left[ F \left( d[\lt x\rt] = 0 \right) \right] \& F \left( d[\lt x\rt] \neq 0 \right) \]
% 05_always_steadystate_abovezero

\[ P \geq 1 \left[ F \left( G \left( [\lt x\rt] > 0 \right) \right) \right] \]
% 05a_finally_active

\[ P \geq 1 \left[ F \left( G \left( [\lt x\rt] > 0 \& d[\lt x\rt] = 0 \right) \right) \right] \]
% 05b_finally_active_steadystate

\[ P \geq 1 \left[ G \left( F \left( [\lt x\rt] > 0 \right) \right) \right] \]
% 05c_always_active_again
P>=1 [ F( G( [<<x>>] = 0 )) ]
% 06_finally_inactive

P>=1 [ G( d[<<x>>] < 0 ) ]
% 07a_always_decreasing_activity
P>=1 [ G( d[<<x>>] <= 0 ) & not G(d[<<x>>] = 0) ]
% 07b_always_decreasing_activityWeakly

P>=1 [ G( d[<<x>>] > 0 ) ]
% 08a_always_increasing_activity
P>=1 [ G( d[<<x>>] >= 0 ) & not G(d[<<x>>] = 0) ]
% 08b_always_increasing_activityWeakly

P>=1 [ F( d[<<x>>]>0 ) & ( d[<<x>>]>0 U ( G d[<<x>>] < 0 ))) ]
% 09a_activity_peaks_and_falls
P>=1 [ not G(d[<<x>>] = 0) & F( d[<<x>>]>0 ) &
( d[<<x>>]>0 U ( not G(d[<<x>>] = 0) & G(d[<<x>>]>=0)))) ]
% 09b_activity_peaks_and_fallsWeakly

P>=1 [ F( d[<<x>>]<0 ) & ( d[<<x>>]<0 U ( G d[<<x>>] > 0 ))) ]
% 10a_activity_falls_and_rises
P>=1 [ not G(d[<<x>>] = 0) & F( d[<<x>>]<0 ) &
( d[<<x>>]<0 U ( not G(d[<<x>>] = 0) & G(d[<<x>>]>=0)))) ]
% 10b_activity_falls_and_risesWeakly

P>=1 [ F( d[<<x>>]>0 ) & ( d[<<x>>]>0 U ( F( d[<<x>>]<0 ) &
G( d[<<x>>] < 0 U G( d[<<x>>]==0)))) ]
% 11_activity_peaks_and_falls_then_steadystate
P>=1 [ F( d[<<x>>]<0 ) & ( d[<<x>>]<0 U ( F( d[<<x>>]>0 ) &
G( d[<<x>>] > 0 U G( d[<<x>>]==0)))) ]
% 12_activity_falls_and_rises_then_steadystate

P>=1 [ G( [<<x>>] <= 0.0001) & not G( [<<x>>] = 0 ) ]
% 14a_rare_events_0.0001
P>=1 [ G( [<<x>>] <= 0.001) & not G( [<<x>>] = 0 ) ]
% 14b_rare_events_0.001
P>=1 [ G( [<<x>>] <= 0.01) & not G( [<<x>>] = 0 ) ]
% 14c_rare_events_0.01
P>=1 [ G( [<<x>>] <= 0.1) & not G( [<<x>>] = 0 ) ]
% 14d_rare_events_0.1
P>=1 [ G( [<<x>>] <= 1) & not G( [<<x>>] = 0 ) ]
% 14e_rare_events_1
P>=1 [ G( [<<x>>] <= 10) & not G( [<<x>>] = 0 ) ]
P>=1 [ (time>0 & time <=250 ) -> G ( [<<x>>] <= 0.0001 ) & not G ( [<<x>>] = 0 ) ]
% 14a_time_rare_0001
P>=1 [ (time>250 & time <=500 ) -> G ( [<<x>>] <= 0.0001 ) & not G ( [<<x>>] = 0 ) ]
% 14b_time_rare_0001
P>=1 [ (time>500 & time <=750 ) -> G ( [<<x>>] <= 0.0001 ) & not G ( [<<x>>] = 0 ) ]
% 14c_time_rare_0001
P>=1 [ (time>750 & time <=1000 ) -> G ( [<<x>>] <= 0.0001 ) & not G ( [<<x>>] = 0 ) ]
% 14d_time_rare_0001
P>=1 [ (time>0 & time <=250 ) -> G ( [<<x>>] <= 0.001 ) & not G ( [<<x>>] = 0 ) ]
% 14a_time_rare_001
P>=1 [ (time>250 & time <=500 ) -> G ( [<<x>>] <= 0.001 ) & not G ( [<<x>>] = 0 ) ]
% 14b_time_rare_001
P>=1 [ (time>500 & time <=750 ) -> G ( [<<x>>] <= 0.001 ) & not G ( [<<x>>] = 0 ) ]
% 14c_time_rare_001
P>=1 [ (time>750 & time <=1000 ) -> G ( [<<x>>] <= 0.001 ) & not G ( [<<x>>] = 0 ) ]
% 14d_time_rare_001
P>=1 [ (time>0 & time <=250 ) -> G ( [<<x>>] <= 0.01 ) & not G ( [<<x>>] = 0 ) ]
% 14a_time_rare_1
P>=1 [ (time>250 & time <=500 ) -> G ( [<<x>>] <= 0.01 ) & not G ( [<<x>>] = 0 ) ]
% 14b_time_rare_1
P>=1 [ (time>500 & time <=750 ) -> G ( [<<x>>] <= 0.01 ) & not G ( [<<x>>] = 0 ) ]
% 14c_time_rare_1
P>=1 [ (time>750 & time <=1000 ) -> G ( [<<x>>] <= 0.01 ) & not G ( [<<x>>] = 0 ) ]
% 14d_time_rare_1
P>=1 [ (time>0 & time <=250 ) -> G ( [<<x>>] <= 1 ) & not G ( [<<x>>] = 0 ) ]
% 14a_time_rare_one
P>=1 [ (time>250 & time <=500 ) -> G ( [<<x>>] <= 1 ) & not G ( [<<x>>] = 0 ) ]
% 14b_time_rare_one
P>=1 [ (time>500 & time <=750 ) -> G ( [<<x>>] <= 1 ) & not G ( [<<x>>] = 0 ) ]
% 14c_time_rare_one
P>=1 [ (time>750 & time <=1000 ) -> G ( [<<x>>] <= 1 ) & not G ( [<<x>>] = 0 ) ]
% 14d_time_rare_one
P>=1 [ (time>10 ) -> (G ( [<<x>>] > 0)) ]
% 15_always_activity_after_10_timesteps
D.2 Metabolite library

P=1 [ (time>0 & time <=250 ) -> G ( [<<x>>] = 0 ) ]
% 01a_01_0to250
P=1 [ (time>250 & time <=500 ) -> G ( [<<x>>] = 0 ) ]
% 01b_01_250to500
P=1 [ (time>500 & time <=750 ) -> G ( [<<x>>] = 0 ) ]
% 01c_01_500to750
P=1 [ (time>750 & time <=1000 ) -> G ( [<<x>>] = 0 ) ]
% 01d_01_750to1000
P=1 [ (time>900 & time <=1000 ) -> G ( [<<x>>] = 0 ) ]
% 01e_01_900to1000
P=1 [ G ( [<<x>>] = 0 ) ]
% 01_always_steadystate_zero

P=1 [ G ( d[<<x>>] = 0 & [<<x>>]>0 ) ]
% 02_always_steadystate_above_zero
P=1 [ (time>900 & time <=1000 ) -> G ( d[<<x>>] = 0 & [<<x>>]>0 ) ]
% 02a_always_steadystate_above_zero900to1000

P=1 [ G ( d[<<x>>] = 0 ) ]
% 03_always_steadystate_any_value

P=1 [ F ( G ( [<<x>>]=0 & d[<<x>>]=0 ) & F (d[<<x>>] !)= 0 ) ]
% 04_changing_and_finally_steadystate_of_zero
P=1 [ F ( G ( [<<x>>]=0 & d[<<x>>]=0 ) & F (d[<<x>>] !)= 0 ) ]
% 05_changing_and_finally_steadystate_above_zero
P>=1 [ F( G( [<<x>>]=0 ) & F( [<<x>>] != 0 ) ]
% 06_changing_and_finally_steadystate_anyvalue
P>=1 [ G ( [<<x>>] < 0 ) ]
% 07a_decreasing
P>=1 [ G ( [<<x>>] <= 0 ) & not G( [<<x>>] = 0 ) ]
% 07b_decreasing_weakly
P>=1 [ G ( [<<x>>] > 0 ) ]
% 08a_increasing
P>=1 [ G ( [<<x>>] >= 0 ) & not G( [<<x>>] = 0 ) ]
% 08b_increasing_weakly
P>=1 [ F( [<<x>>]>0 ) & ( [<<x>>]>0 U ( G [<<x>>] < 0 )))]
% 09a_peaks_and_falls
P>=1 [ not G([<<x>>] = 0 ) & F( [<<x>>]>0 ) &
    ( [<<x>>]<0 U ( not G([<<x>>] = 0 )& G([<<x>>]<=0))) ]
% 09b_peaks_and_falls_weakly
P>=1 [ F( [<<x>>]<0 ) & ( [<<x>>]<0 U ( G [<<x>>] > 0 )))]
% 10a_falls_and_rises
P>=1 [ not G([<<x>>] = 0 ) & F( [<<x>>]<0 ) &
    ( [<<x>>]<=0 U ( not G([<<x>>] = 0 )& G([<<x>>]>=0))) ]
% 10b_falls_and_rises_weakly
P>=1 [ F( [<<x>>]>0 ) & ( [<<x>>]>0 U ( F( [<<x>>]<0 ) &
    G ( [<<x>>] < 0 U G ( [<<x>>]=0 )))]) ]
% 11_peaks_and_falls_then_steadystate
P>=1 [ F( [<<x>>]<0 ) & ( [<<x>>]<0 U ( F( [<<x>>]>0 ) &
    G ( [<<x>>] > 0 U G ( [<<x>>]=0 )))]) ]
% 12_falls_and_rises_then_steadystate
P>=1 [ (F( [<<x>>] != 0 ) & not( F( G( [<<x>>]=0 & [<<x>>]=0 ) ))) ]
% 13_activity_and_not_finally_steadystate_of_zero
P>=1 [ ( [<<x>>] = 0 ) ]
% 15_always_activity
P>=1 [ G ( [<<x>>] <= 510 & [<<x>>] >= 490 ) ]
% 15b_always_around_500
P>=1 [ G ( [<<x>>] <= 510 & [<<x>>] >= 500 ) ]
% 15c1_always_just_above_500
P>=1 [ G ( [<<x>>] <= 500 & [<<x>>] >= 490 ) ]
% 15c2_always_just_below_500
P>=1 [ G ( [<<x>>] >= 0.9*max[<<x>>] ) & not G ( [<<x>>] = 0 ) ]
% 15d_always_at_least_90percent_of_max_and_greaterthan_0
P>=1 [ (time>=900 & time <=1000 ) ->
   G ( [<<x>>] >= 0.95*max[<<x>>] ) & not G ( [<<x>>] = 0 ) ]
% 15e_always_at_least_95percent_of_max_and_greaterthan_0_900to1000
P>=1 [ G ( [<<x>>] >= 0.5*max[<<x>>] ) &
       not G ( [<<x>>] = 0 )){[<<x>>] >= 0.5*max[<<x>>]}]
% 15f_after_max_always_at_least_90percent_of_max_and_greaterthan_0
P>=1 [ G ( [<<x>>] >= 0.9*max[<<x>>] ) & not G ( [<<x>>] = 0 )){[<<x>>] > 400}]
% 15f_gt_400_always_at_least_90percent_of_max_and_greaterthan_0
P>=1 [ G ( [<<x>>] >= 0.85*1.5 & [<<x>>] <= 1.15*1.5 ){[<<x>>] = 1.5 }]
% 15g_15
P>=1 [ G ( [<<x>>] <= 0.0001 ) & not G ( [<<x>>] = 0 ) ]
% 14a_always_low_concentrations_0.0001
P>=1 [ G ( [<<x>>] <= 0.001 ) & not G ( [<<x>>] = 0 ) ]
% 14b_always_low_concentrations_0.001
P>=1 [ G ( [<<x>>] <= 0.01 ) & not G ( [<<x>>] = 0 ) ]
% 14c_always_low_concentrations_0.01
P>=1 [ G ( [<<x>>] <= 0.1 ) & not G ( [<<x>>] = 0 ) ]
% 14d_always_low_concentrations_0.1
P>=1 [ G ( [<<x>>] <= 1 ) & not G ( [<<x>>] = 0 ) ]
% 14e_always_low_concentrations_1
P>=1 [ G ( [<<x>>] <= 10 ) & not G ( [<<x>>] = 0 ) ]
% 14f_always_low_concentrations_10
Fig. 1. Comparison of average reaction activity for selected subsystems in the min-growth model (red) and enhanced-growth model (blue); the activity of subsystems in the enhanced-growth model has significantly improved compared to the min-growth model. Continued in Fig. 2.
Fig. 2. (Continued from Fig. 1). Comparison of average reaction activity for selected subsystems in the min-growth model (red) and enhanced-growth model (blue); the activity of subsystems in the enhanced-growth model has significantly improved compared to the min-growth model, apart from Inorganic Ion Transport and Metabolism. That subsystem comprises only two reactions whose activities, as far as we can see from this analysis, are not influenced by the growth conditions.
Fig. 3. Hierarchical clustering of the subsystems Euclidean distance for (a) min-growth model, behaviour of averaged activity traces, (b) enhanced-growth model, behaviour of averaged activity traces, (repeated from Figure ?? in the main paper), (c) enhanced-growth model, structural relatedness per subsystem.
References

1. M Heiner, D Gilbert, and R Donaldson. Petri nets in systems and synthetic biology. In *Schools on Formal Methods (SFM)*, pages 215–264. LNCS 5016, Springer, 2008.
2. R Heinrich and S Schuster. *The regulation of cellular systems*. Ch.3. NewYork: Chapman&Hall, pp.75-111, 1996.
3. M Heiner, M Schwarick, and J Wegener. Charlie – an extensible Petri net analysis tool. In *Proc. Petri Nets*, volume 9115 of *LNCS*, pages 200–211. Springer, June 2015.
4. M Schwarick. *MARCIE- User Manual*. Computer Science Department BTU Cottbus-Senftenberg, TR-2016-02, 2016.