

# PB050 Modelování a predikce v SB

## Nástroj BioCHAM

zpracováno pro verzi 3.7

14.1.2021

*David Šafránek*

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.



# Nástroj BioCHAM 3

<http://lifeware.inria.fr/biocham3/>

oproti COPASI možnost kvalitativní analýzy

kvantitativní analýza rozšířena o možnost  
ověřování dynamických vlastností

analýza robustnosti modelu vzhledem k dané  
vlastnosti

rozšířené možnosti specifikace modelu

# BioCHAM Model

rule-based přístup k zápisu modelu  
struktura modelu:

**sada pravidel** (vzorce kinetiky nepovinné)  
specifikace iniciálních podmínek  
parametry, makra, invarianty, události  
deklarace molekul a kompartmentalizace  
temporální vlastnosti

# BioCHAM Model – Objects

Konstrukce pro reprezentaci substrátů - species

*object* = *molecule* | *molecule::cname* (makro)molekula  
kvalifikace kompartmentem nepovinná  
bez udání se uvažuje „Default“

*molecule* =  
| *name* název molekuly  
| *molecule-molecule* komplex (asociativní, komutativní operace)  
| *molecule~{sname,...,sname}* modifikace (stav) molekuly  
| *#name* genetický materiál (promotor, gen)  
| ( *molecule* )

*Příklady:* *S1, S2, E, E-S1, E-S1-S2*  
*(E-S1)~{p1,p2}, E-S1~{p1}*  
*TF-#genA*

# BioCHAM Model – Objects

Konstrukce pro reprezentaci substrátů – patterny

*object\_pattern* = *molecule\_pattern* | *molecule\_pattern*::*simple\_pattern*

*variable* = ? | *\$name*

*simple\_pattern* = *name* | *variable*

*molecule\_pattern* = *simple\_pattern*  
| *molecule\_pattern*-*molecule\_pattern*  
| *molecule\_pattern*~{*simple\_pattern*,...,*simple\_pattern*}  
| *molecule\_pattern*~*variable*  
| *gene*  
| ( *molecule\_pattern* )

*Příklady:* *declare* *MAPK*~{ {}, {*p1*}, {*p1*,*p2*} }  
*MAPK*~{*p1*, ?} ... *MAPK*~{*p1*}, *MAPK*~{*p1*,*p2*}  
*MAPK*~*\$P* ... *MAPK*~{ }, *MAPK*~{*p1*}, *MAPK*~{*p1*,*p2*}  
*MAPK*-? ... jakýkoliv komplex obsahující *MAPK*

*Ize navíc omezit:*

*MAPK*~*\$P* where *p2* not in *\$P*.

# BioCHAM Model - Rules

Konstrukce pro reprezentaci reakcí

*reaction* = *kinetics* **for** *basic\_reaction*

| *basic\_reaction*

| *name* : *basic\_reaction*

| *name* : *kinetics* **for** *basic\_reaction*

anotace reakce vzorcem kinetiky

deklarace reakce

pojmenování reakce (nepovinné)

*basic\_reaction* = *solution* => *solution*.

| *solution* =[*object*]=> *solution*.

| *solution* <=> *solution*.

| *solution* <=[*object*]=> *solution*.

ireversibilní reakce

modulovaná ireversibilní reakce

reversibilní reakce

modulovaná reversibilní reakce

*solution* = \_

| *object*

| *integer*\**object*

| *solution* + *solution*

| ( *solution* )

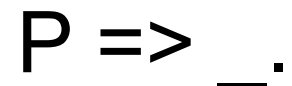
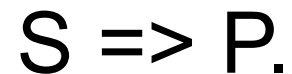
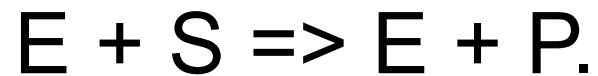
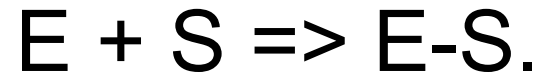
prázdný objekt

stechiometrie

komplexace

# BioCHAM Model - Rules

Příklady



# BioCHAM Model - Rules

Anotace reakcí matematickými vzorci

```
kinetics = simple_kin  
  | (simple_kin , simple_kin)  
  
simple_kin = [object]  
  | float  
  | simple_kin 'op' simple_kin  
  | MA(simple_kin)  
  | MM(simple_kin, simple_kin)  
  | H(simple_kin, simple_kin, int)  
  | (simple_kin)
```

deklarace vzorce  
deklarace vzorce pro reversibilní reakci

koncentrace objektu  
konstanta  
alg. výraz ('op' -- \*, +, -, /, ^)  
makro pro mass action kinetiku  
makro pro Michaelis-Menten kinetiku  
makro pro Hillovu kinetiku



# BioCHAM Model - Rules

Příklady:

- 1)  $0.1*[S]*[E]$  for  $E + S \Rightarrow E-S$ .  
 $1*[E-S]$  for  $E-S \Rightarrow E + S$ .  
 $1*[E-S]$  for  $E-S \Rightarrow E + P$ .
- 2)  $(0.1*[S]*[E], 1*[E-S])$  for  $E + S \Leftrightarrow E-S$ .
- 3)  $(MA(0.1), MA(1))$  for  $E + S \Leftrightarrow E-S$ .  
 $MA(1)$  for  $E-S \Rightarrow E + P$ .
- 4)  $MM(100, 22)$  for  $S = [E] \Rightarrow P$ .
- 5)  $(MA(3.3), MA(0.4))$  for  $MEK \sim \$P + RAF \sim \{p1\} \Leftrightarrow MEK \sim \$P - RAF \sim \{p1\}$   
where  $p2$  not in  $\$P$ .

# BioCHAM Model - Rules

Reprezentace kinetiky mass action

makro  $MA(k)$  je expandováno dle zákona mass action:

$MA(k)$  for  $A + B \Rightarrow C$ .

$MA(k) == k*[A]*[B]$

$MA(k)$  for  $A \Rightarrow B$ .

$MA(k) == k*[A]$

# BioCHAM Model - Rules

Reprezentace kinetiky Michaelis-Menten a Hillovy kinetiky

makro MM( V, K ) je expandováno:

MM( V, K ) for S => P.

$$\text{MM}( V, K ) == V * [S] / (K + [S])$$

makro H( m, K, n ) je expandováno:

H( m, K, n ) for S => P.

$$\text{H}( m, K, n ) == m * [S]^n / (K^n + [S]^n)$$

# BioCHAM Model - Rules

Parametrizace modelu

libovolnou hodnotu v kinetických výrazech lze parameterizovat definicí parametru:

**parameter( *name*, *float* ).**

**Příklad:**

**parameter( *k*, *0.1* ).**  
 **$k \cdot [A] \cdot [B]$  for  $A + B \Rightarrow C$ .**

# BioCHAM Model - Rules

Iniciální podmínky

*přímé nastavení, nastavení kopií:*

present( *object\_pattern*, *value* ).  
present( *object\_pattern*, *name* ).

*nastavení na nulu:*

absent(object\_pattern).

# BioCHAM Model - Cvičení

```
>load_biocham(enzyme_ma).  
>numerical_simulation(200).  
>plot.
```

```
>load_biocham(enzyme_mm).  
>numerical_simulation(200).  
>plot.
```

```
>list_molecules.  
>list_initial_state.  
>list_parameters.  
>list_rules.
```

# BioCHAM – Kvantitativní LTL

Nechť  $AP$  je množina atomických propozic. Formule  $\varphi$  je formulí *lineární temporální logiky (LTL)* pokud splňuje následující:

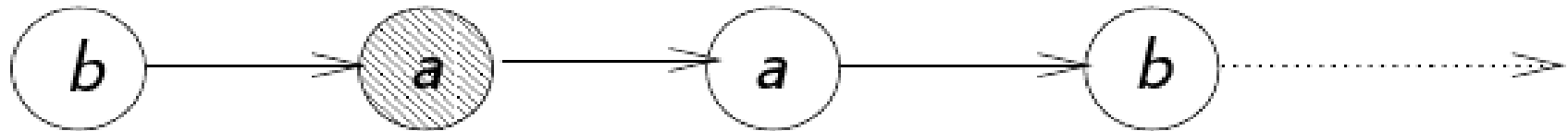
- $\varphi = p$  pro libovolné  $p \in AP$
- Jsou-li  $\varphi_1$  a  $\varphi_2$  formule LTL, pak:
  - $\neg\varphi_1$ ,  $\varphi_1 \wedge \varphi_2$  a  $\varphi_1 \vee \varphi_2$  jsou formule LTL
  - **X** $\varphi_1$ , **F** $\varphi_1$  a **G** $\varphi_1$  jsou formule LTL
  - $\varphi_1$  **U**  $\varphi_2$  je formule LTL

Jako atomické propozice jsou uvažovány logické výrazy nad reálnými hodnotami (koncentracemi), značeno LTL(R).

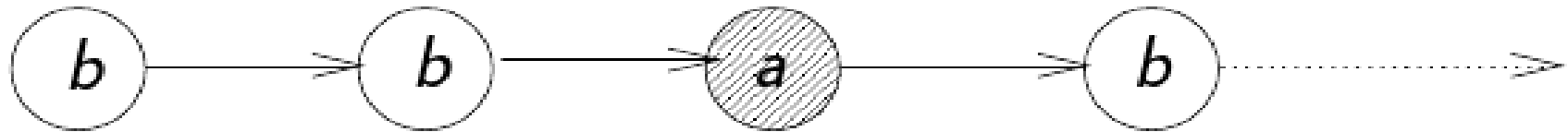
# BioCHAM – Kvantitativní LTL

Sémantika

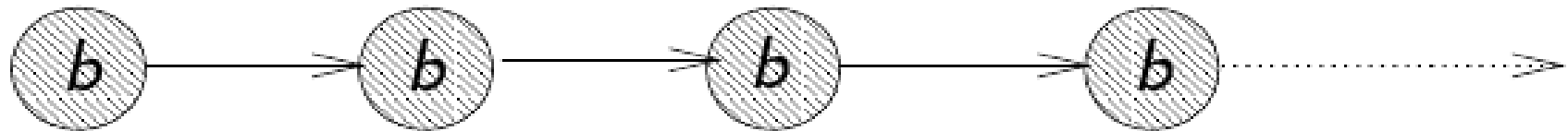
$Xa$



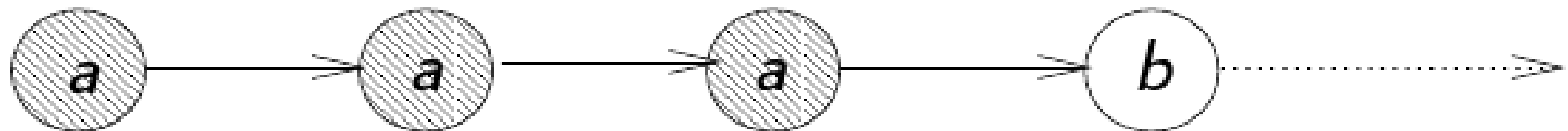
$Fa$



$Gb$



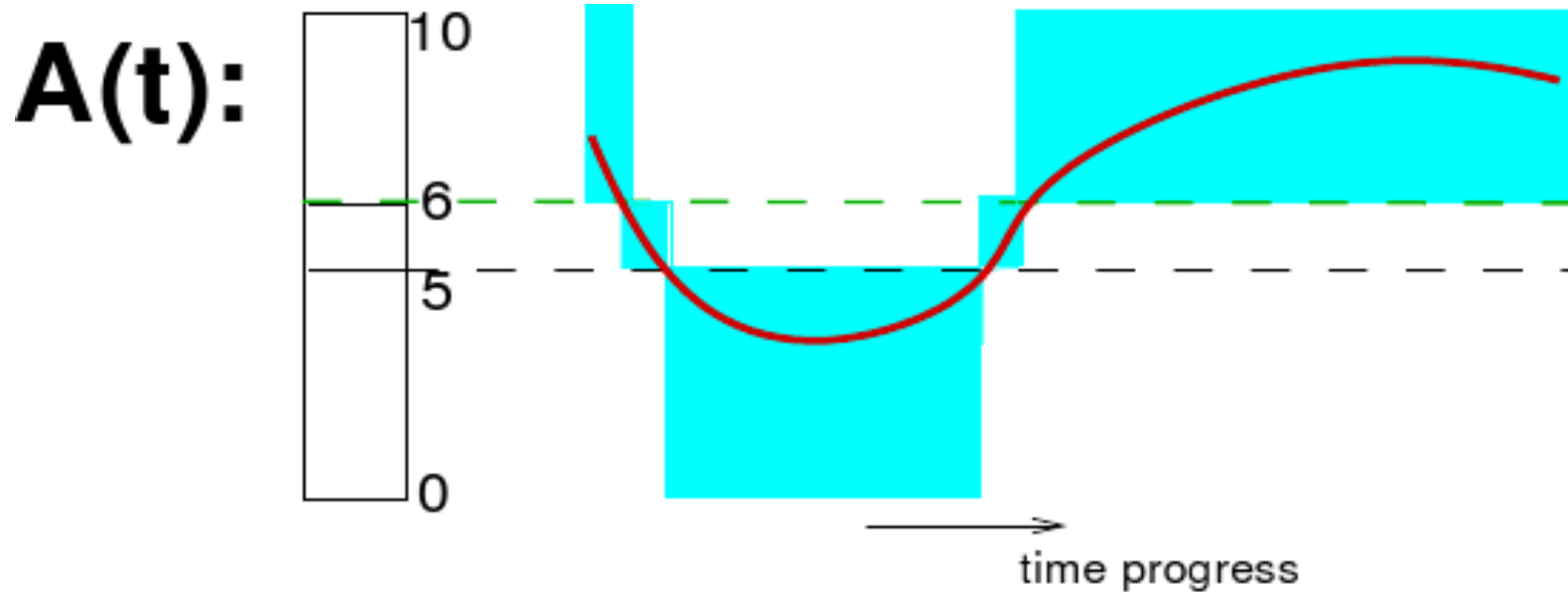
$aUb$





# BioCHAM – Kvantitativní LTL

Interpretace nad (konečnými) simulacemi



**F** ( $[A] < 5$ )

**G** ( $[A] \geq 0 \ \& \ [A] \leq 10$ )

$[A] > 6 \ \mathbf{U} \ (\mathbf{X}(\ ([A] \leq 6 \ \& \ [A] \geq 5) \ \mathbf{U} \ (\mathbf{X}([A] < 5 \ \mathbf{U} \ (\mathbf{X}(\ ([A] \geq 5 \ \& \ [A] \leq 6) \ \mathbf{U} \ [A] > 6))))))$

# BioCHAM – Kvantitativní LTL

Formální sémantika pro (konečné) simulace

Uvažujeme simulaci (trace)  $T=(s_0, s_1, \dots, s_n)$ . Každý bod  $s_i$  je určen trojicí zahrnující aktuální čas, hodnoty simulovaných proměnných a hodnoty jejich derivací v daném bodě.

Definujeme relaci splnitelnosti  $T \models \phi$  pro atomické propozice triviálně, dále podle struktury formule:

- $s_i \models \phi \wedge \psi$  iff  $s_i \models \phi$  and  $s_i \models \psi$ ,
- $s_i \models \phi \vee \psi$  iff  $s_i \models \phi$  or  $s_i \models \psi$ ,
- $s_i \models \mathbf{X}\phi$  iff  $i < n$  and  $s_{i+1} \models \phi$ , or  $i = n$  and  $s_n \models \phi$ ,
- $s_i \models \mathbf{F}\phi$  iff  $\exists j \in [i, n]$  such that  $s_j \models \phi$ ,
- $s_i \models \mathbf{G}\phi$  iff  $\forall j \in [i, n], s_j \models \phi$ ,
- $s_i \models \phi \mathbf{U} \psi$  iff  $\exists j \in [i, n]$  s. t.  $s_j \models \psi$  and  $\forall k \in [i, j-1], s_k \models \phi$ .

# BioCHAM – Kvantitativní LTL

## Syntax LTL(R)

$ltl$  = condition

|  $(ltl)$

|  $F(query)$

|  $G(query)$

|  $X(query)$

|  $ltlUtl$

|  $!(ltl)$

|  $ltl \& ltl$

|  $ltl | ltl$

|  $ltl \text{ xor } ltl$

|  $ltl \rightarrow ltl$

|  $ltl \leftrightarrow ltl$

finally

globally

next

until

negation

conjunction

disjunction

exclusive or

implication

equivalence

# BioCHAM – Kvantitativní LTL

Syntax LTL(R)

*condition = simple\_kin < simple\_kin*  
| *simple\_kin > simple\_kin*  
| *simple\_kin = simple\_kin*  
| *simple\_kin =< simple\_kin*  
| *simple\_kin >= simple\_kin*

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – ověřování vlastnosti vzhledem k simulaci

```
>load_biocham(enzyme_ma).
```

```
>numerical_simulation(200).
```

```
>check_ltl(G(d([P])/dt>=0)).
```

```
>check_ltl(([E]>=90) U (X([E]>10 & [E]<90) U (X ([E]>=90)))).
```

```
>check_ltl((d([E])/dt>=0) U (X((d([E])/dt<0) U (X (d([E])/dt>=0)))).
```

*Příkaz check\_ltl se provádí na poslední provedené simulaci.*

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – vlastnosti vyjadřující se k časovému bodu

```
>check_ltl(G(Time=10 -> [P]<100) & (Time=100 -> [P]>200)).
```

```
>check_ltl(G(Time>100 -> G([P]>470))).
```

```
>check_ltl(G((Time>5 & Time<20)->(d([E-S])/dt<1 & d([E-S])/dt>(-1)))).
```

# BioCHAM – Šablony formulí

Parametrizace konstant v atomických propozicích

Uvažujme formuli  $\Phi = \mathbf{F}([A]>7 \ \& \ \mathbf{F}([A]<3))$ .

Z formule lze vytvořit šablonu parametrizací konstant:

$$\Phi(x,y) = \mathbf{F}([A]>x \ \& \ \mathbf{F}([A]<y))$$

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – valuace parametrizovaných formulí

>solve(F(G([P]=v))).

>solve(F(G([P]>=v))).

>solve(G(Time>v -> G([P]>470))).

>solve(G(Time>100 -> G([P]>v))).

>solve(G((Time>5 & Time<v)->(d([E-S])/dt<1 & d([E-S])/dt>(-1)))).

>solve(F(Time=v & ([P]>=100 & [P]=<110))).

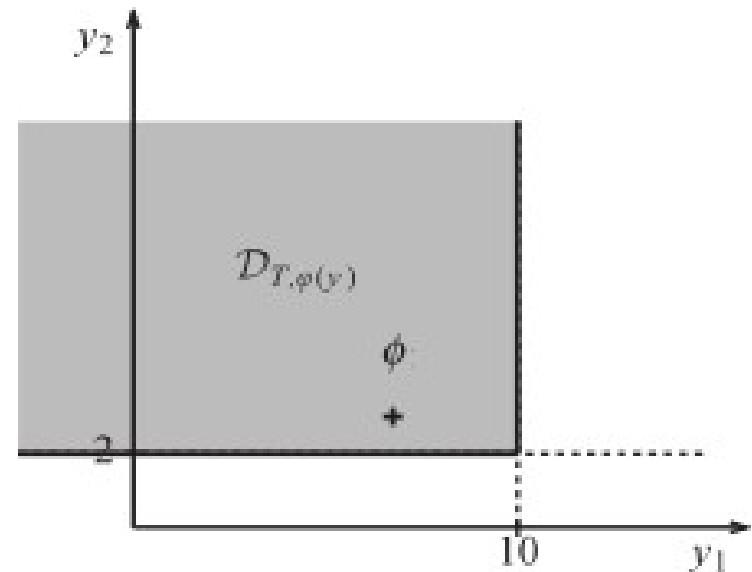
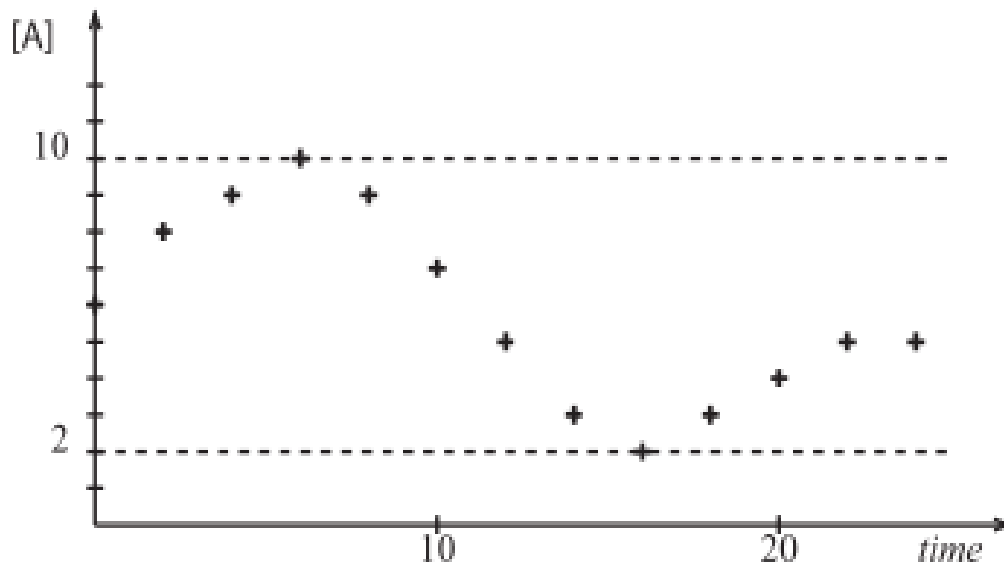
*Funkce solve pracuje nad poslední simulací.*



# BioCHAM – Šablony formulí

Parametrizace konstant v atomických propozicích

Platnost šablony  $\Phi(y_1, y_2) = \mathbf{F}([A] > y_1 \ \& \ \mathbf{F}([A] < y_2))$  pro danou simulaci T je vymezena podprostorem v  $\mathbf{R}^2$ , tzv. **doménou validity**, zn.  $D_{T, \phi(x, y)}$ , definovanou všemi hodnotami proměnných  $y_1, y_2$ , pro něž je formule splněna.



# BioCHAM – Šablony formulí

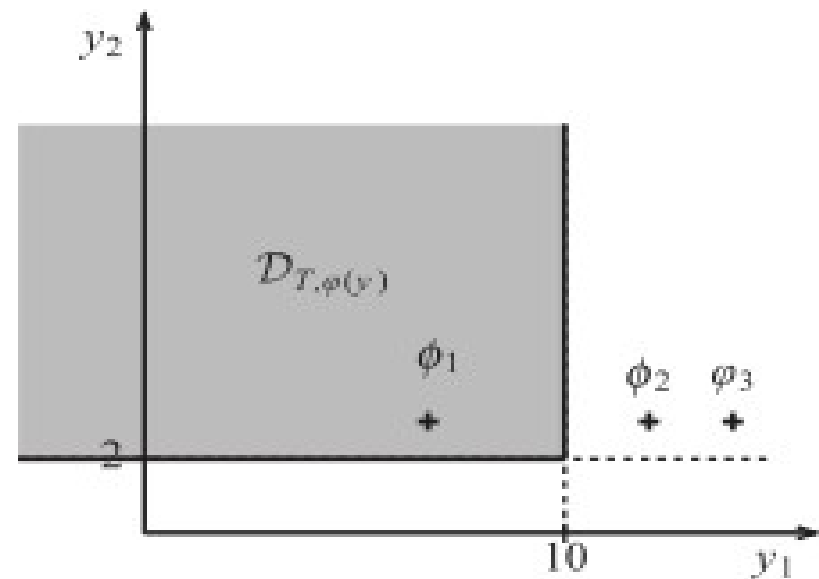
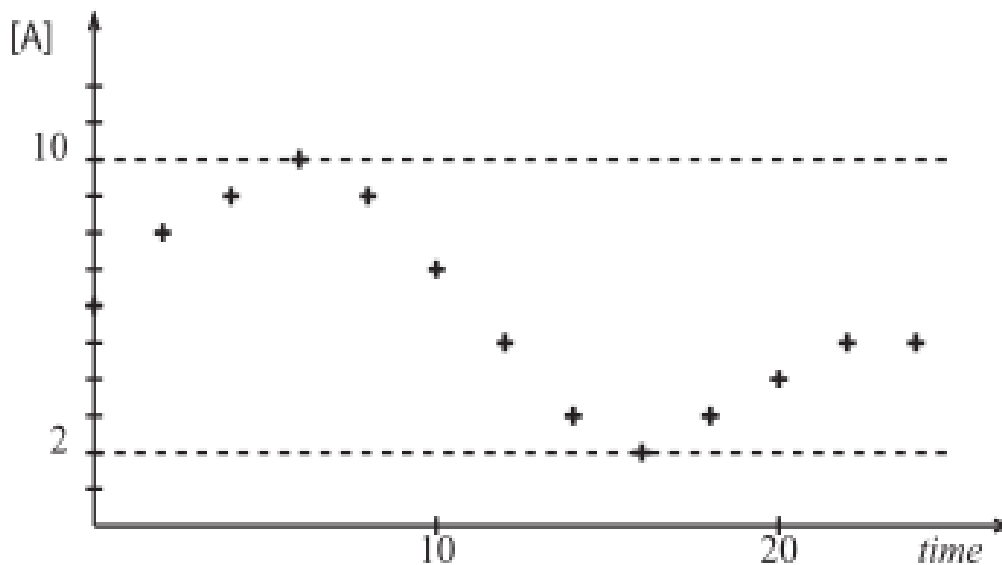
Parametrizace konstant v atomických propozicích

Pro  $\Phi(y_1, y_2) = \mathbf{F}([A] > y_1 \ \& \ \mathbf{F}([A] < y_2))$  uvažujme instance:

$$\Phi_1 = \mathbf{F}([A] > 7 \ \& \ \mathbf{F}([A] < 3))$$

$$\Phi_2 = \mathbf{F}([A] > 12 \ \& \ \mathbf{F}([A] < 3))$$

$$\Phi_3 = \mathbf{F}([A] > 14 \ \& \ \mathbf{F}([A] < 3))$$



Formule  $\Phi_1$  a  $\Phi_2$  leží mimo doménu validity, jsou od ní různě (Euklidovsky) vzdáleny.

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – stupeň porušení vlastnosti

Pro danou simulaci  $T$  a formuli  $\Phi$  definujeme **stupeň porušení** (violation degree):

$$vd(T, \Phi) = \text{dist}(\Phi, D_{T, \Phi(v)})$$

kde  $v$  je vektor všech proměnných ve formuli  $\Phi$

- hodnota kvantizující, jak daleko je formule od platnosti pro trajektorii  $T$
- pokud  $vd(T, \Phi) = 0$ , formule je splněna

Příklad:

$$vd(T, \Phi_1) = 0, \quad vd(T, \Phi_2) = 2, \quad vd(T, \Phi_3) = 4$$

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – stupeň porušení vlastnosti

Normalizovaná forma, tzv. **stupeň splněnosti** (satisfaction degree):

$$sd(T, \phi) = \frac{1}{1 + vd(T, \phi)} \in [0, 1],$$

- hodnota charakterizující platnost formule pro trajektorii T:  
TRUE odpovídá hodnotě 1  
FALSE je určeno asymptoticky (k 0)

Příklad:

$$sd(T, \Phi_1) = 1, \quad sd(T, \Phi_2) = 0.3, \quad sd(T, \Phi_3) = 0.2$$

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – míra nesplnění vlastnosti

`>satisfaction_degree(F(G([P]>=v)), [v], [470], 200).`

- vrátí hodnotu stupně splnění
- příkaz inherentně provádí simulace
- délka simulace určená posl. parametrem (zde 200)
- v tomto případě je hodnota parametru 470 a vlastnost je v modelu `enzyme_ma` splněna, tedy míra nesplnění bude 1

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – míra nesplnění vlastnosti

>satisfaction\_degree(F(G([P]>=v)),[v],[470],200).

>satisfaction\_degree(F(G([P]>=v)),[v],[100],200).

>satisfaction\_degree(F(G([P]>=v)),[v],[800],200).

# BioCHAM – Robustnost

Míra robustnosti

Uvažujme systém  $S$ , vlastnost  $a$ , a množinu vychýlení systému  $P$  (tzv. ***množinu perturbací***). Např. změna parametrů.

Pro charakterizaci faktu, do jaké míry systém zachovává požadovanou vlastnost při uplatnění perturbací, je využíván pojem ***robustnosti*** (definovaný H. Kitanem).

Robustnost je definována pomocí ***evaluační funkce***  $D_a^S$  popisující míru zachování požadované vlastnosti systému při perturbaci.

Uvažujeme-li pro každou perturbaci  $p$  z  $P$  pravděpodobnost uplatnění  $prob(p)$ , je robustnost definována jako vážený průměr hodnot evaluační funkce přes množinu perturbací:

$$R_{a,P}^S = \int_{p \in P} prob(p) D_a^S dp$$

# BioCHAM – Robustnost

Použití LTL(R) – míra robustnosti LTL vlastností

Uvažujme vlastnost  $\Phi = F(G([P] \geq v))$  požadující ustálení koncentrace produktu  $[P]$  na hodnotě alespoň  $v$ .

Uvažujme vychýlení kinetických parametrů modelu v daném rozsahu (množina perturbací). Jaký vliv má na platnost uvažované vlastnosti  $\Phi$ ?

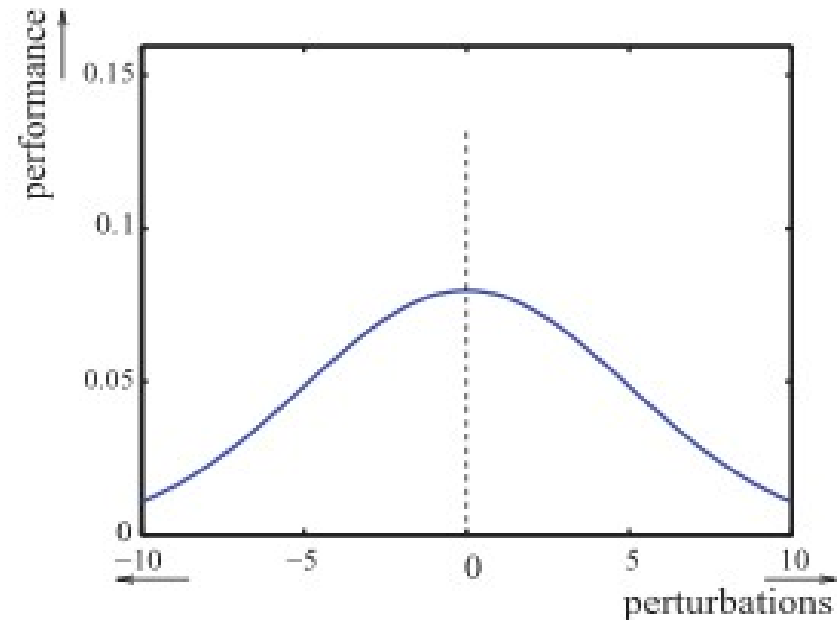
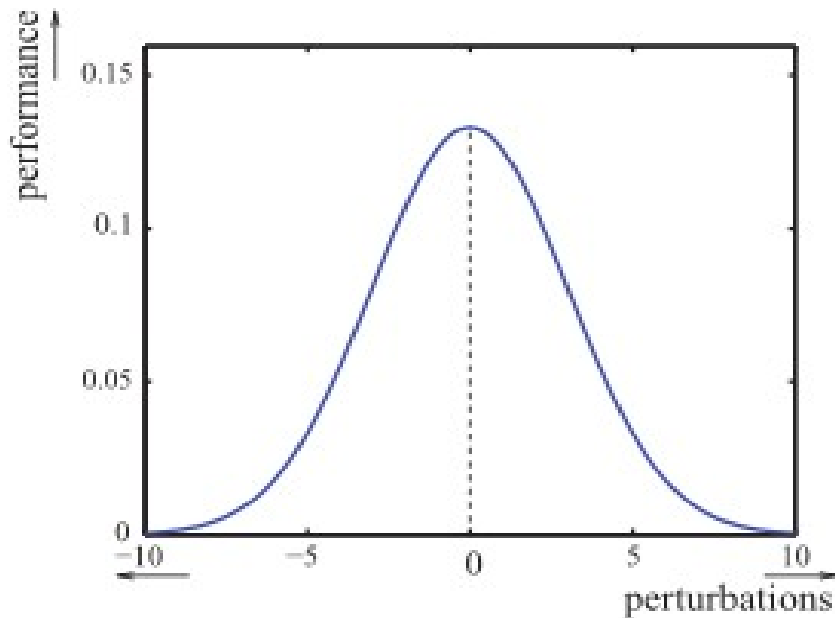
Definujeme pojem **robustnosti vlastnosti  $\Phi$  vůči množině perturbací  $P$**  jako vážený průměr stupně splnění na simulacích získaných pro tyto perturbace (hodnota z  $[0,1]$ ):

$$R_{\Phi, P}^S = \int_{p \in P} \text{prob}(p) \text{sd}(T_p, \Phi) dp,$$



# BioCHAM – Kvantitativní LTL

Použití LTL(R) – absolutní vs. relativní robustnost



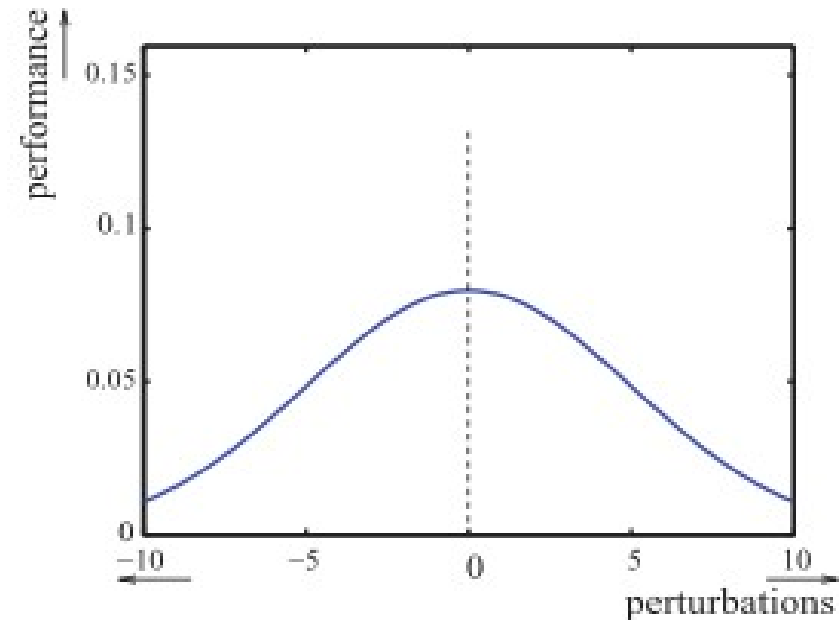
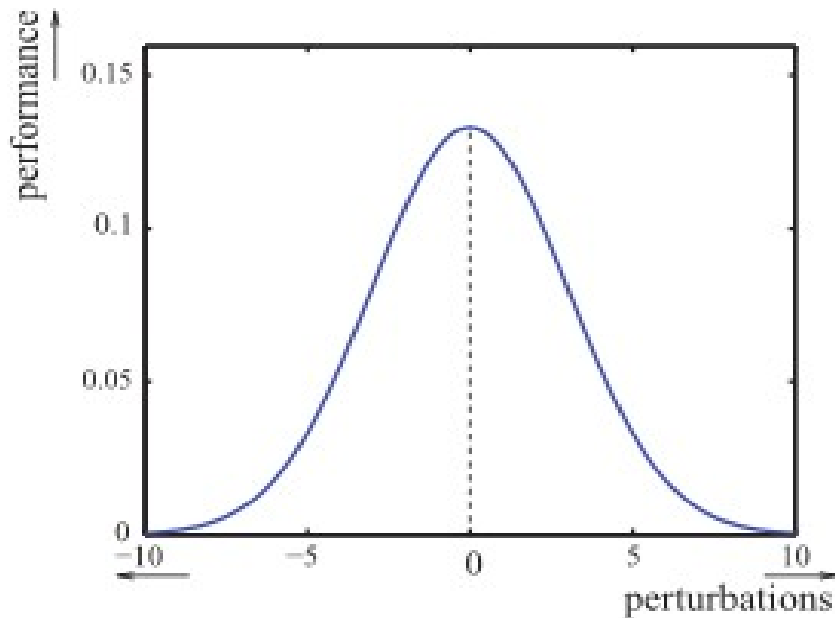
Shodná robustnost vlastnosti „performance“ dvou různých systémů při téže množině uniformě rozložených perturbací.

Uvažovaná robustnost postihuje průměrné chování při uniformě rozložených perturbacích (v rozsahu  $[-10,10]$ ). Pokud uvažujeme perturbaci 0 (nominální chování), vlastnost obou systémů se liší, ačkoliv robustnosti (obsah pod křivkou) jsou totožné.

Z tohoto důvodu hovoříme o **absolutní robustnosti**.

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – absolutní vs. relativní robustnost



Shodná robustnost vlastnosti „performance“ dvou různých systémů při téže množině uniformě rozložených perturbací.

**Relativní robustnost** je definována poměřením absolutní robustnosti vůči hodnotě evaluační funkce nominálního případu (zn.  $p^*$ ):

$$R_{\phi, P}^{S, p^*} = R_{\phi, P}^S / sd(T_{p^*}, \phi)$$

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – míra robustnosti

```
>robustness_normal([k1],[0.5],F(G([P]>=v)),[v],[470],50,200).
```

Uvažován perturbační model pro vychýlení parametru  $k_1$  v obou směrech dle normálního rozložení s koeficientem variance 0.5 okolo původní nastavené hodnoty.

Příkaz provede 50 simulací dle různého náhodného samplování  $k_1$ , délka každé simulace je 200.

Vrátí absolutní robustnost danou přímo samplováním  $k_1$  nebo relativní robustnost (poměřenou vůči neperturovanému  $k_1$ ).

Absolutní robustnost je počítána jako průměrný stupeň splnění vlastnosti přes všechny samplované parametrizace.

Relativní robustnost je absolutní robustnost poměřená stupněm splnění vlastnosti v neperturovaném modelu.

# BioCHAM – Kvantitativní LTL

Použití LTL(R) – míra robustnosti

>robustness([k1],[0.5],F(G([P]>=v)),[v],[470],50,200).

>robustness([k2],[0.5],F(G([P]>=v)),[v],[470],50,200).

>robustness([k3],[0.5],F(G([P]>=v)),[v],[470],50,200).

>robustness([k1,k2,k3],[0.5,0.5,0.5],F(G([P]>=v)),[v],[470],50,200).

>robustness([k1,k2,k3],[0.5,0.5,0.5],F(G([P]>=v)),[v],[100],50,200).

>robustness([k1,k2,k3],[0.5,0.5,0.5],F(G([P]>=v)),[v],[800],50,200).

# Literatura k BioCHAMu

Dokumentace viz

<http://contraintes.inria.fr/BIOCHAM3/DOC/manual.html>

Přednáška připravena dle článku:

Aurélien Rizk, Grégory Batt, François Fages, Sylvain Soliman: A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics* 25(12): (2009)