

# Langages formels pour la biologie systémique dans la machine abstraite biochimique BIOCHAM

Sylvain Soliman

INRIA Paris-Rocquencourt  
Équipe CONTRAINTES – F. Fages  
FP6 TEMPO (INSERM) – AgroBI INSIGHT (INRA)

Avancées en Sciences de l'Information  
9 octobre 2007 – Académie des sciences

# Plan de l'exposé

- 1 Introduction
  - Biologie Systémique
  - Modélisation
  - Approche Langage
- 2 BIOCHAM
  - Description
  - Un langage pour les règles d'interaction
  - Un langage pour les propriétés biologiques
  - Automatisation
- 3 Conclusion

# Biologie Systémique

## H. Kitano, ICSB 2000

“Systems Biology aims at systems-level understanding [which] requires a set of principles and methodologies that links the behaviors of molecules to systems characteristics and functions.”

- **Analyse** des données (post-)génomiques produites par les technologies à “haut débit” (intégrées dans des bases de données telles GO, KEGG, BioCyc, etc.) ;
- **Intégration** de données hétérogènes sur un problème spécifique ;
- **Compréhension et Prédiction** des comportements ou des interactions dans de grands réseaux de gènes ou de protéines.

*cf. l'introduction du cours d'immunologie systémique du professeur Kourilsky au Collège de France*



## Des approches variées

Une grande variété d'outils développés pour des **problèmes spécifiques**, et selon différentes approches de modélisation :

- Statistique ;
- Discrète (booléen ou à niveaux) ;
- Continue, équations différentielles (EDO ou EDP) ;
- Stochastique ;
- Hybride ;
- ...

Concepts qui ont fait leurs preuves en informatique  $\Rightarrow$  **échange et réutilisation** de modèles (et modules) existants formalisés (SBML) ; **raisonnement** multi-échelle, compositionnel, topologique, etc. sur le comportement global du système.

# Approche Langage

## Modèles **qualitatifs** : des *diagrammes* aux

- Réseaux booléens [Thomas 73]
- $\pi$ -calcul [Regev et al. 99-01, Nagasaki et al. 00]
- Pathway logic [Eker et al. 02]
- Systèmes de transition concurrents, BIOCHAM [Chabrier-Fages 03]
- Bio-ambients [Regev et al. 03]

## Modèles **quantitatifs** : des systèmes *d'équations différentielles* aux

- Réseaux de Petri hybrides [Hofstadt-Thelen 98, Matsuno et al. 00]
- Automates hybrides [Alur et al. 01, Ghosh-Tomlin 01]
- Hybrid CC [Bockmayr-Courtois 01]
- BIOCHAM [Chabrier-Fages-Soliman 04]

## Approche Langage (2)

Modèles **compartmentés** : des *schémas* aux

- Automates cellulaires
- MemBrane calculus [Cardelli et al. 03, Danos 03]
- Bio-ambients [Regev et al. 03]
- BIOCHAM

Modèles **stochastiques** : des simulations à la *Gillespie* (1976-2003)  
aux

- $\pi$ -calcul stochastique [Cardelli 04]
- BIOCHAM

Mais qu'en est-il des **propriétés du système** (i.e. les données expérimentales) ?

# La Machine Abstraite Biochimique

## “Qu'est-ce que BIOCHAM ?”

- Un environnement de modélisation ;
- Un langage formel pour la description des interactions (moléculaires) et des **résultats d'expériences** sur le système biologique étudié ;
- Un outil d'analyse avec trois niveaux d'abstraction : booléen, stochastique, différentiel.

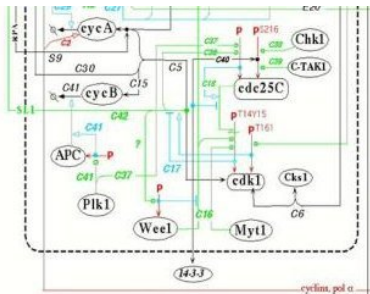
## “Que modélise-t-on ?”

- principalement : des réseaux d'interactions entre protéines (et gènes) avec leur dynamique, au niveau intracellulaire ;
- expérimentalement : des populations de cellules, des tissus, . . .

## Exemple : le cycle cellulaire d'après Kohn

Extrait d'une transcription (formelle, donc **non ambiguë et analysable**) de la carte du contrôle du cycle cellulaire des mammifères de Kurt Kohn [MBC 1999].

Un ensemble d'environ 800 règles avec 500 composés.



$cdk1\{p1,p2,p3\} + cycA \Rightarrow cdk1\{p1,p2,p3\}-cycA.$

$cdk1\{p1,p2,p3\} + cycB \Rightarrow cdk1\{p1,p2,p3\}-cycB.$

...

$cdk1\{p1,p3\}-cycA = [ Wee1 ] \Rightarrow cdk1\{p1,p2,p3\}-cycA.$

$cdk1\{p1,p3\}-cycB = [ Wee1 ] \Rightarrow cdk1\{p1,p2,p3\}-cycB.$

...

$cdk1\{p2,p3\}-cycA = [ Myt1 ] \Rightarrow cdk1\{p1,p2,p3\}-cycA.$

$cdk1\{p2,p3\}-cycB = [ Myt1 ] \Rightarrow cdk1\{p1,p2,p3\}-cycB.$

...

$cdk1\{p1,p2,p3\} = [ cdc25C\{p1,p2\} ] \Rightarrow cdk1\{p1,p3\}.$

$cdk1\{p1,p2,p3\}-cycA = [ cdc25C\{p1,p2\} ] \Rightarrow cdk1\{p1,p3\}-cycA.$

$cdk1\{p1,p2,p3\}-cycB = [ cdc25C\{p1,p2\} ] \Rightarrow cdk1\{p1,p3\}-cycB.$

...

$\_ = [ E2F13-DP12-gE2 ] \Rightarrow cycA.$

...

$cycB = [ APC\{p1\} ] \Rightarrow \_.$

...

# Expressions Cinétiques et Stochastiques

Quand l'information est disponible, chaque règle d'interaction peut être équipée avec une **expression arithmétique** interprétée comme un taux de réaction, i.e. une vitesse ou une probabilité de réaction.

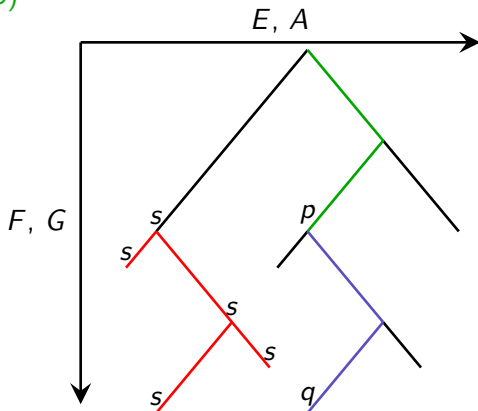
```
present(MPF~{p}, 1).
absent({MPF, cdc25C~{p1,p2}, Wee1, Wee1~{p}, APC}).
...
parameter(k2u, 3).
parameter(k1cc, 0.25).
...
k1cc                                for _ => MPF~{p}.
MA(k2u)                              for MPF = [ APC ] => _ .
MA(k2u)                              for MPF~{p} = [ APC ] => _ .
...
MA(k3cc)                             for MPF~{p} = [ cdc25C~{p1,p2} ] => MPF .
MA(k4cc)                             for MPF = [ Wee1 ] => MPF~{p}.
...
```

Ces expressions sont ignorées au niveau booléen.

# Formaliser les propriétés biologiques du système en CTL

(Computation Tree Logic [voir par exemple Clarke 99])

- activation, **accessibilité**  $EF(p)$
- à propos des "chemins"
  - produit intermédiaire  
 $EF(p \wedge EF(q))$
  - **checkpoint**  
 $\neg E(\neg p U q)$
- stabilité, stationnarité
  - état(s) **stable(s)**  
 $s \Rightarrow AG(s)$
  - **oscillations**  
 $EG(EF(\neg t) \wedge EF(t))$



# Vérification de modèle sur la carte de Kohn avec NuSMV

```
biocham: check_reachable(cdk46~{p1,p2}-cycD~{p1}).  
Ei(EF(cdk46~{p1,p2}-cycD~{p1})) is true  
  
biocham: check_checkpoint(cdc25C~{p1,p2}, cdk1~{p1,p3}-cycB).  
Ai(!(E(!(cdc25C~{p1,p2}) U cdk1~{p1,p3}-cycB))) is true  
  
biocham: nusmv(Ai(AG(!(cdk1~{p1,p2,p3}-cycB) -> checkpoint(Wee1, cdk1~{p1,p2,p3}-cycB))))).  
Ai(AG(!(cdk1~{p1,p2,p3}-cycB)->!(E(!(Wee1) U cdk1~{p1,p2,p3}-cycB)))) is false  
  
biocham: why.  
-- Loop starts here  
  cycB-cdk1~{p1,p2,p3} is present  
  cdk7 is present  
  cycH is present  
  cdk1 is present  
  Myt1 is present  
  cdc25C~{p1,p2} is present  
  
rule_114   cycB-cdk1~{p1,p2,p3}=[cdc25C~{p1,p2}]=>cycB-cdk1~{p2,p3}.  
  cycB-cdk1~{p2,p3} is present  
  cycB-cdk1~{p1,p2,p3} is absent  
  
rule_74   cycB-cdk1~{p2,p3}=[Myt1]>cycB-cdk1~{p1,p2,p3}.  
  cycB-cdk1~{p2,p3} is absent  
  cycB-cdk1~{p1,p2,p3} is present
```

Moins d'une minute (mais toujours bien plus difficile que des circuits électroniques de même taille).

# LTL avec contraintes numériques et PCTL

LTL est une restriction de CTL\* aux opérateurs  $F$  et  $G$ .

$$F([MPF] > 0.2 \ \& \ G(d([APC])/dt < 0))$$
$$period(MPF, 22)$$

LTL et PCTL (Probabilistic CTL) permettent de formaliser les propriétés de simulations stochastiques.

$$P(EF(cdk46 \sim \{p1, p2\} - cycD \sim \{p1\})) > 0.5$$

LTL et PCTL sont vérifiées par *model-checking* (sur des traces ou des ensembles de simulations).

# Propriétés biologiques spécifiées en logique temporelle

```
add_specs({
    reachable(MPF~{p}),
    reachable(MPF),
    reachable(cdc25C),
    reachable(cdc25C~{p1,p2}),
    reachable(Wee1),
    reachable(APC),
    ...
    loop(MPF~{p},MPF),
    oscil(cdc25C),
    oscil(cdc25C~{p1,p2}),
    loop(Wee1,Wee1~{p}),
    oscil(CKI),
    oscil(APC),
    ...
    Ai(EG((!(MPF)) ->checkpoint(cdc25C~{p1,p2},MPF)))
}).
```

# Recherche automatique de règles d'interactions

Supposons que l'activation de MPF par la forme active de Cdc25C manque dans le modèle. La vérification de la spécification donne :

```
biocham: check_all.
```

The specification is not satisfied.

This formula is the first not verified:  $E_i(EF(MPF))$

Corrections suggérées par “theory revision” :

```
biocham: learn_one_rule(elementary_interaction_rules).
```

Rules tested: 4218

Good rules to be added: 3

```
_=[cdc25C~{p1,p2}]=>MPF
```

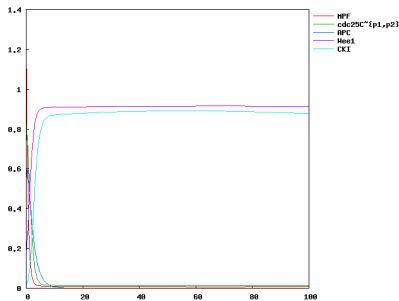
```
MPF~{p}=[cdc25C~{p1,p2}]=>MPF
```

```
CKI+MPF~{p}=[cdc25C~{p1,p2}]=>CKI-MPF
```

# Recherche automatique de valeurs de paramètres

Prenons une cinétique en Loi d'Action de Masse pour la règle nouvellement trouvée :

```
MA(k3cc) for MPF~{p}=[cdc25C~{p1,p2}]=>MPF.  
parameter(k3cc,0.1).
```

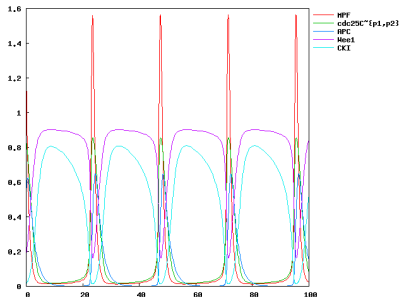


Le comportement du modèle est loin des oscillations observées dans le cycle cellulaire.

## Recherche automatique de valeurs de paramètres (2)

Valeurs satisfaisantes suggérées par une recherche :

```
learn_param([k3cc],[0,5],20,osci(MPF,4)&F([MPF]<0.05),100).
```



Found parameters that make `osci(MPF,4) & F([MPF]<0.05)` true:  
`parameter(k3cc,2.5)`.

# Conclusion

- Un langage simple pour **décrire les processus biologiques**
  - sémantique booléenne
  - sémantique stochastique
  - sémantique différentielle
- Un langage pour **décrire les propriétés biologiques** du système
  - logique temporelle CTL
  - logique temporelle LTL avec contraintes numériques
- Un environnement original d'aide à la modélisation
  - **recherche de règles d'interaction** (à partir d'une spécification CTL)
  - **estimation de paramètres** (à partir d'une spécification LTL)

# Trois Défis pour la Biologie Systémique Computationnelle

- Modèles
  - Composition/décomposition/réutilisation : modularité ;
  - Modèles multi-cellulaires multi-échelle ;
  - Abstractions formelles pour les réseaux d'influence/interaction.
- Outils pour raisonner sur les modèles
  - Validation de modèles avant publication (SBML) ;
  - Inférence de règles d'interaction/paramètres ;
  - Découverte de cibles (potentielles) de médicaments.
- Enseignement de la Biologie
  - Modèles et outils formels au centre des cours.

## BIOCHAM

