

TP : modélisation à l'aide d'un réseau de Petri de la régulation du gène *fls2*

Cette régulation ainsi que les connaissances biologiques décrites ci-dessous sont fictives et ont été imaginées pour réaliser ce TP.

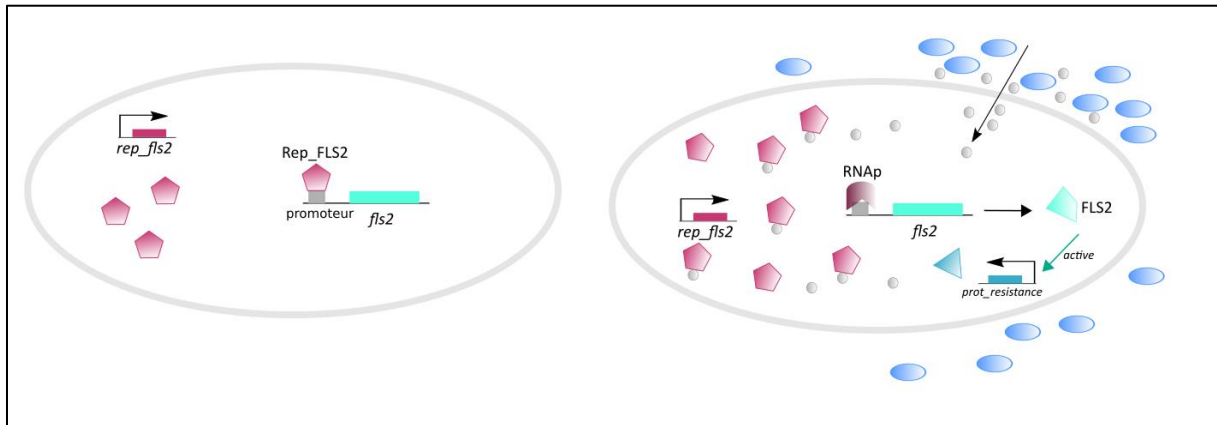


Figure 1 : description schématique de la régulation du gène *fls2* en absence de stress (gauche) et en présence de stress (droite). Les bactéries sont représentées par des ovales bleus et les molécules de toxines par des ronds gris clair. Les gènes sont représentés par des rectangles de couleur précédés de leur promoteur (flèche, excepté pour le gène *fls2* où le promoteur est identifié par un carré gris). Le même code couleur a été conservé pour représenter les produits protéiques des gènes.

En absence de stress (absence de bactérie) la transcription du gène *fls2* est réprimée par le répresseur Rep_FLS2 dont la transcription est constitutive (pas régulée par le réseau).

En présence de bactéries, celles-ci produisent des toxines qui pénètrent dans la cellule.

- Ces toxines sont reconnues par Rep_FLS2 qui les séquestre en formant un complexe.
- Les protéines Rep_FLS2 étant engagées dans la formation du complexe avec les toxines, libèrent le promoteur de *fls2* le rendant accessible à la RNA polymérase (RNAP), qui en se liant au promoteur, va permettre la transcription du gène et la synthèse de la protéine FLS2.
- FLS2 va à son tour activer la synthèse des protéines impliquées dans la résistance de la plante à l'attaque des bactéries.
- Les toxines libres (non liées à Rep_FLS2) sont dégradées.
- Tous les complexes peuvent se dissocier (Rep_FLS2_toxine, Rep_FLS2_promoteur, RNAP_promoteur).
- Les protéines Rep_FLS2 et FLS2 sont dégradées

Simplifications et hypothèses pour la modélisation :

- On ne modélisera pas la synthèse de la RNA polymérase qui est supposée être en nombre de molécules suffisantes (100 molécules à l'état initial)
- On modélisera la synthèse de Rep_FLS2 en considérant qu'elle n'est pas régulée par le système et qu'à l'état initiale 30 protéines sont présentes

- On fusionnera les étapes de transcription et de traduction en une seule étape de synthèse qui en partant du gène donnera la protéine correspondante
- Le promoteur du gène *fls2* existera sous trois formes : libre, liée à Rep_FLS2 ou lié à RNAP

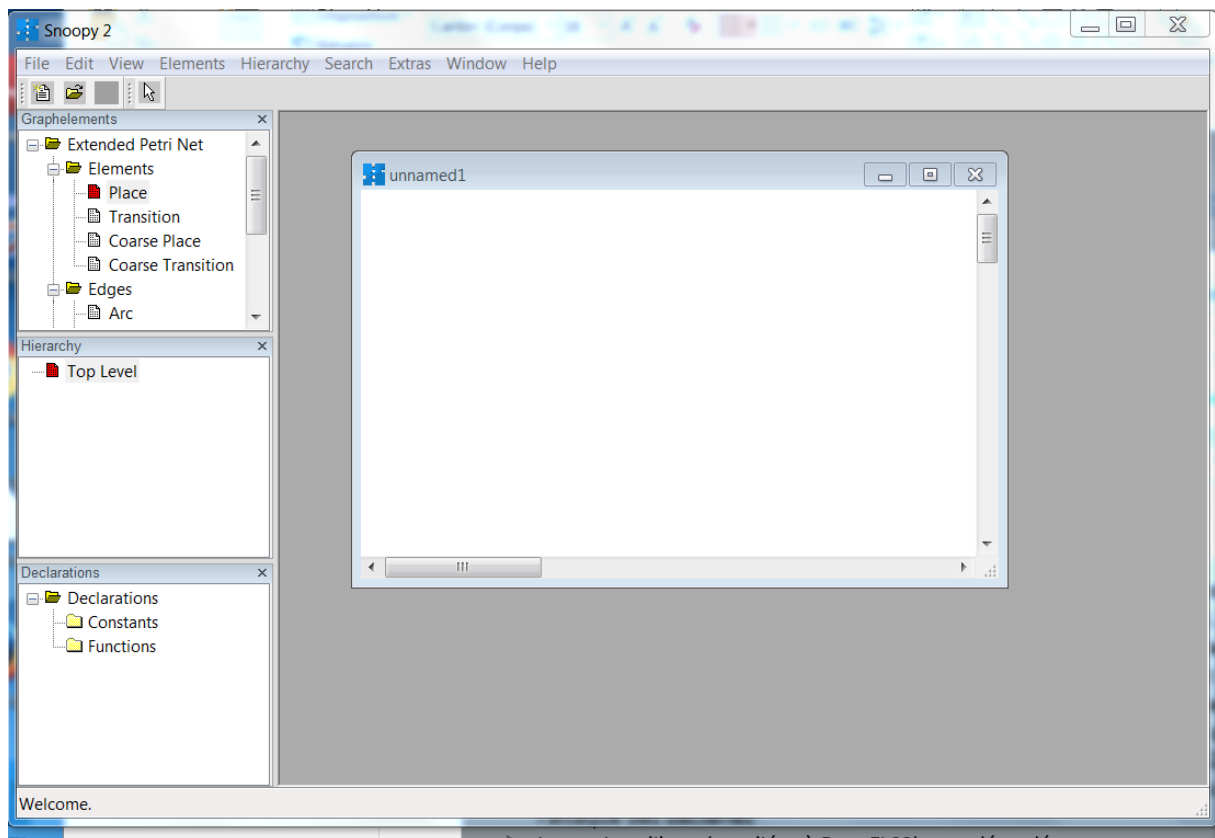
Questions :

1. Enumérer les composants moléculaires du processus qui devront être pris en compte dans la modélisation et qui constitueront les places du réseau
2. Donner la liste des réactions qui constitueront les transitions
3. Définir les conditions initiales du système (nombre de jetons dans les places à l'état initial)
4. Créer le réseau de Petri correspondant en utilisant le logiciel Snoopy.
5. Animer votre réseau pour suivre le « token game »

Utilisation de Snoopy pour la création du réseau :

Créer un nouveau modèle et choisir « Extended Petri Net » ce qui permettra de modéliser des arcs test et inhibiteur.

On obtient l'environnement suivant :



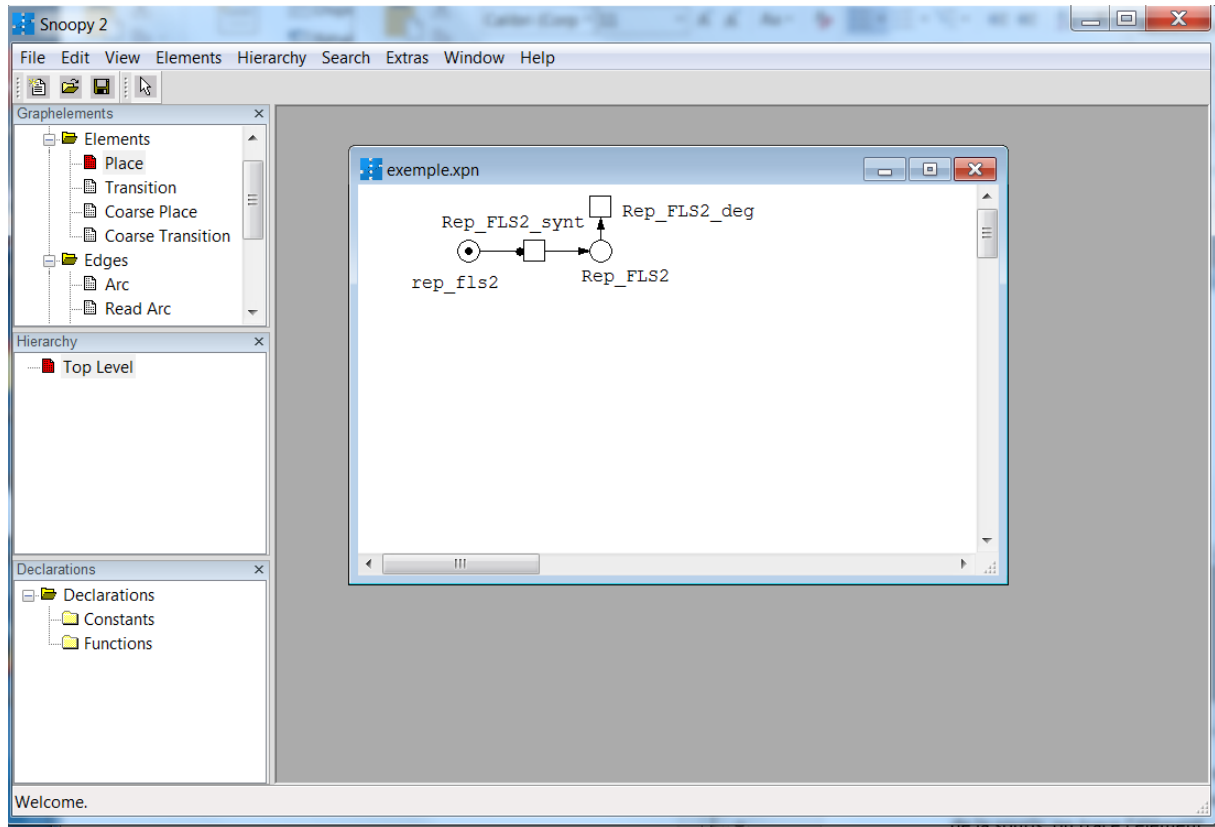
On choisit l'élément du réseau que l'on veut dessiner dans le menu à gauche (Place, Transition, Arc, Read Arc etc..). On se place ensuite dans la fenêtre « blanche » et en cliquant avec le bouton gauche de la souris, on trace l'élément.

En double cliquant sur l'objet tracé :

- Place : donner un nom et indiquer le nombre de jetons présents dans la place à l'état initial du système (Marking)

- Transition : donner un nom
- Arc : donner le poids si différent de 1 (Multiplicity)

Pour sauvegarder votre travail aller dans File, Save as et donner un nom. Le fichier aura l'extension .xpn qui indique que ce fichier contient un réseau de Petri standard étendu créé par Snoopy.



Quand votre réseau est dessiné, pour lancer le l'animation (le « token game ») :

Dans View, **Start_Anim_Mode** et cliquer sur **Play Forward**

Dans Option :

Single : 1 seule transition réalisable choisie au hasard parmi les transitions réalisables fait feu

Intermediate : un sous-groupe arbitraire de transitions concurrentes réalisables est choisi et les transitions font feu

Maximum : un groupe maximal de transitions réalisables concurrentes est choisi et les transitions font feu

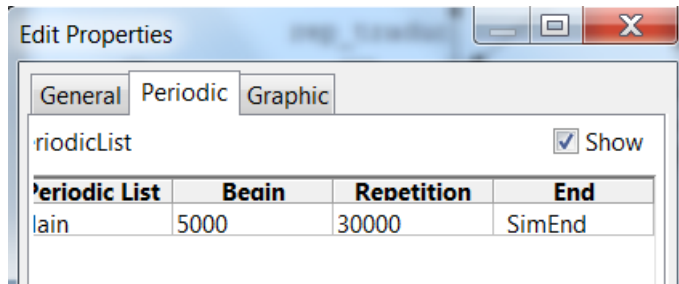
Exporter le réseau sous forme de réseau stochastique :

Aller dans File, choisir Export et cocher Export to Stochastic Petri Net. Un nouveau fichier avec l'extension .spn va être créé indiquant maintenant que le fichier contient un réseau stochastique et non plus standard.

Ouvrir ce nouveau fichier. Il va falloir apporter quelques modifications à notre réseau :

Nous allons programmer l'entrée des toxines bactériennes par pulses. Nous allons modéliser deux pulses, l'un intervenant après 5000 unités de temps et l'autre à 30000 (on fera une simulation sur 60000 unités de temps).

Pour cela : supprimer la transition `external_toxine_supply` et en créer une nouvelle qui sera de type Scheduled Transition. Double cliquer sur la transition, donner le nom puis aller l'onglet periodic et le remplir comme suit :



Mettre une multiplicité de 10000 sur l'arc qui relie la transition à la place `toxin_bact`.

Nous allons aussi rajouter une transition qui sort de la place `proteine_resistance` que nous appellerons `resistance_activation` et l'arc reliant la place `proteine_resistance` à cette transition aura une multiplicité de 2. En effet, sinon les protéines vont s'accumuler au cours du temps car elles ne seront impliquées dans aucune action. Donc si nous voulons voir la cinétique de la mise en place de la résistance par la plante ainsi que celle de son arrêt, il faut rajouter cette transition

Préparer les données pour faire la simulation

Il faut aussi fournir des paramètres pour calculer le délai de déclenchement de chaque transition réalisable.

Créer le tableau contenant ces valeurs.

Aller dans la partie Declarations et cliquer sur Constant.

Dans la fenêtre (qui est vide au départ), faire add constant et remplir la ligne comme indiqué dans la figure ci-dessous. Dans Group et Type, on a un menu déroulant. Faire les choix indiqués. Puis recommencer l'opération pour ajouter les 13 constantes.

Constant Definition

	Show	Constant	Group	Type	Comment	Main
1	<input type="checkbox"/>	c1	parameter	double	rep svnt	0.1
2	<input type="checkbox"/>	c2	parameter	double	Rep FLS2 dea	0.002
3	<input type="checkbox"/>	c3	parameter	double	Prom Rep FLS2 liaison	1
4	<input type="checkbox"/>	c4	parameter	double	Prom Rep FLS2 dissoc	0.01
5	<input type="checkbox"/>	c5	parameter	double	Prom RNAP liaison	0.1
6	<input type="checkbox"/>	c6	parameter	double	Prom RNAP dissoc	0.01
7	<input type="checkbox"/>	c7	parameter	double	fls2 svnt	0.5
8	<input type="checkbox"/>	c8	parameter	double	FLS2 deg	0.001
9	<input type="checkbox"/>	c9	parameter	double	Tox Rep FLS2 liaison	0.005
10	<input type="checkbox"/>	c10	parameter	double	complex dissoc	0.1
11	<input type="checkbox"/>	c11	parameter	double	Prot resistance svntesis	0.1
12	<input type="checkbox"/>	c12	parameter	double	toxine dea	0.0005
13	<input type="checkbox"/>	c13	parameter	double	resistance activation	0.1

Dans View, choisir Start Simulation Mode. Cela ouvre une nouvelle fenêtre. Il reste encore une configuration à réaliser pour indiquer quel paramètre utiliser pour le calcul du délai de déclenchement de chaque transition. Cliquer sur Model Configuration et faire Modify au niveau de Function set. Par défaut, la fonction utilisée pour le calcul est MassAction. On conservera cette fonction. Quand on ouvre pour la première fois la fenêtre, chacune des transitions est indiquée associée à MassAction(1). Pour chaque transition, il va falloir indiquer le nom de la constante qui lui est associée. Le faire comme suit :

Function overview

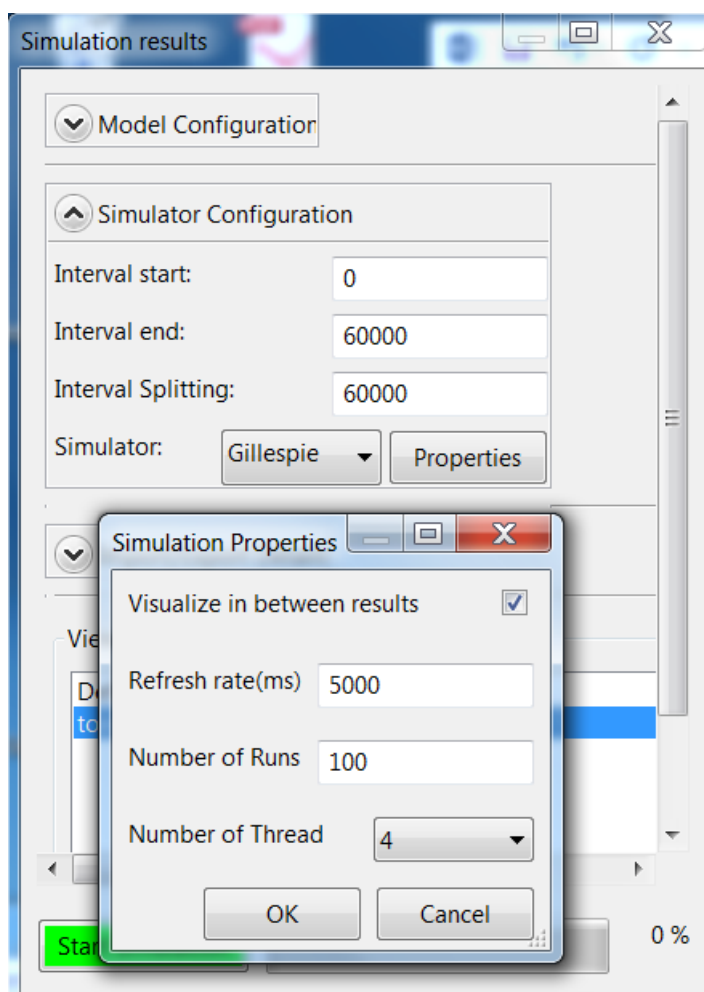
	Main
rep_synt	MassAction(c1)
Rep_FLS2_deg	MassAction(c2)
Prom_Rep_FLS2_liaison	MassAction(c3)
Prom_Rep_FLS2_dissoc	MassAction(c4)
Prom_RNAP_liaison	MassAction(c5)
Prom_RNAP_dissoc	MassAction(c6)
FLS2_deg	MassAction(c8)
Tox_Rep_FLS2_liaison	MassAction(c9)
complex_dissoc	MassAction(c10)
Prot_resistance_synthese	MassAction(c11)
toxine_deg	MassAction(c12)
resistance_activation	MassAction(c13)
FLS2_synt	MassAction(c7)

Vérifiez que le Schedule set contient bien la description suivante de la transition programmée external_toxine_supply (Modify) sinon, le remplir.

	Main:Beain	Main:Repetition	Main:End
toxine_extern_e_supply	5000	30000	_SimEnd

Maintenant votre modèle est configuré.

Nous allons maintenant configurer le simulateur (Simulator Configuration). Renseigner comme indiqué ci-dessous. Pour avoir la petite fenêtre Simulation Properties cliquer sur Properties à côté de Gillespie. On va exécuter 100 simulations de manière à obtenir un comportement moyen lissé des variations du nombre de chaque composant du système au cours du temps. Nous garderons l'algorithme de simulation Gillespie.



Pour finir et pouvoir avoir les courbes montrant comment les composants évoluent au cours du temps, nous allons créer une nouvelle vue dans View, faire add et donner un nom à la vue. Le nom va apparaître dans la fenêtre du dessus. Cliquer sur le nom et faire Edit Node List, pour choisir les composants que l'on veut suivre.

On va pouvoir lancer la simulation !

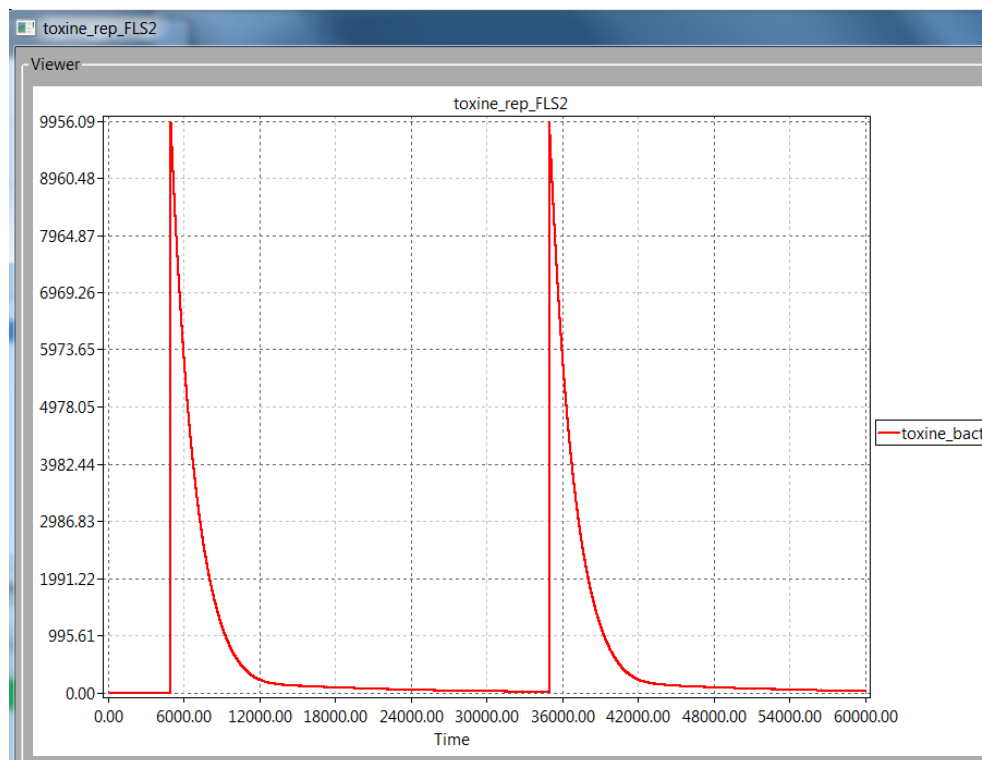
On va d'abord choisir de suivre les toxines bactériennes. Vous devez obtenir le résultat suivant qui indique bien qu'il y a eu deux pulses d'entrée comme programmé à 5000 et un autre 30000 après le premier, soit à 35000 de la simulation.

Ensuite, on va suivre simultanément la cinétique des protéines Rep_FLS2 (répresseur), FLS2 et la protéine de résistance ainsi que la cinétique du complexe Toxine_Rep_FLS2. On ouvre de nouveaux Edit Node List. On enlève la place toxine_bactérienne et on ajoute les 4 autres variables. Faire save.

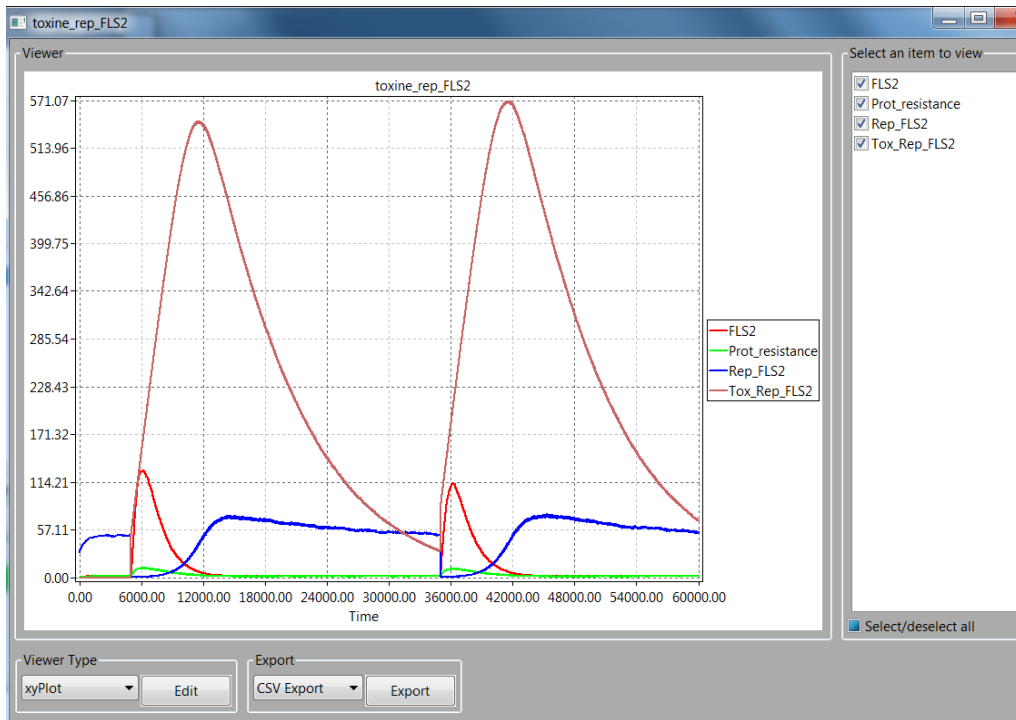
Finalement, on va refaire l'opération et comparer la cinétique du promoteur lié au répresseur et celle du promoteur lié à la RNA polymerase.

Les trois résultats sont donnés dans les figures suivantes. Vérifier que vous obtenez la même chose. Interpréter ces résultats.

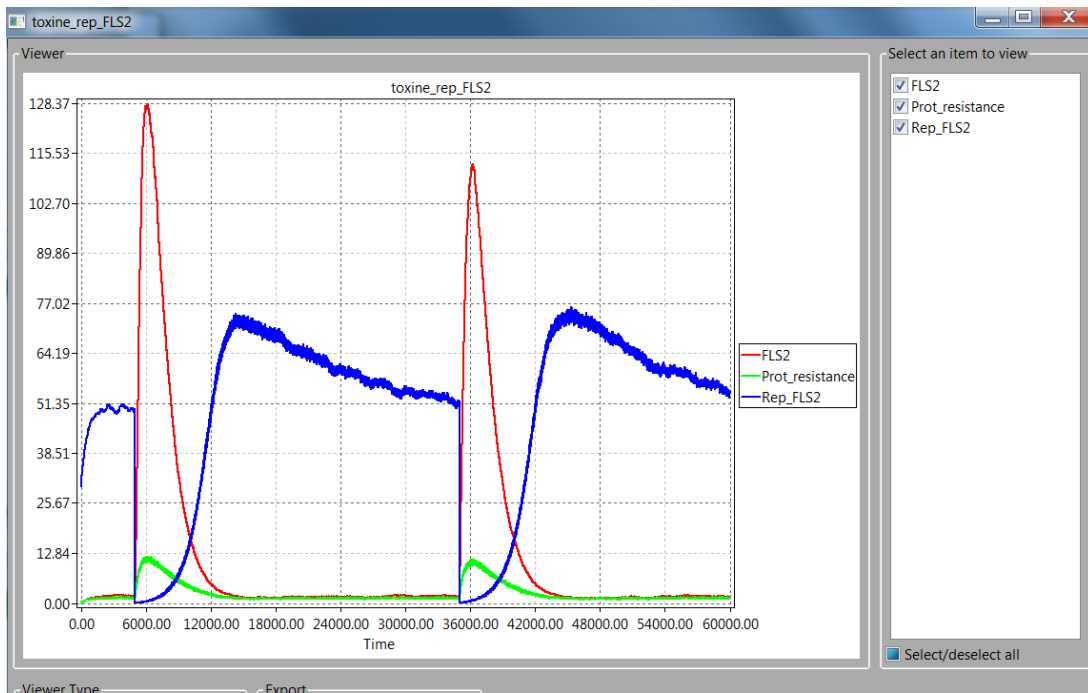
Cinétique des toxines bactériennes :



Cinétique des Protéines et du complexe Toxine-répresseur



Cinétique des protéines sans le complexe



Cinétique des deux états du promoteur

