

Universités Bordeaux 1 et 2
2010 - 2011

Laetitia Bourgeade
Hadrien Mary
Florence Maurier
Jean-Paul Navailles

NuQleosim

Master Biologie Santé
Mention Bioinformatique

Responsable : Marie Beurton Aimar

Abstract

Les cellules eucaryotes ont la particularité d'être subdivisées en structures spécialisées dont le noyau, lieu de stockage de l'information génétique. Cet organite est lui même organisé en trois zones : l'euchromatine, l'hétérochromatine et le nucléole. C'est ce dernier qui est au coeur de ce projet.

Le nucléole, se trouve être le centre de biosynthèse des ribosomes. Son apparence varie en fonction de son activité. Lors d'une biogenèse des ribosomes très active, le nucléole s'organise en trois composantes majeures visibles en microscopie électronique : les centres fibrillaires entourés par le composant fibrillaire dense lui même enchâssé dans le composant granulaire. Le nucléole joue un rôle capital dans l'organisation nucléaire. Ainsi, des mutations au niveau de gènes codant pour des protéines nucléolaires ont été associées à des maladies humaines.

Ce travail présente le développement d'une application (**NuQleoSim**) conçu dans le but d'accompagner les biologistes lors de l'étude du nucléole.

A ce jour, il n'existe encore aucune base de données relative au nucléole. De ce fait, l'un des objectifs de **NuQleoSim** est de proposer une interface de saisie et de consultation des informations disponibles sur le nucléole (images de microscopie, données qualitatives et quantitatives sur ses molécules). Ces informations sont mises en relation avec la partie simulation.

Le nucléole reste difficilement observable, sa modélisation peut donc permettre de mieux comprendre son fonctionnement et d'apporter des pistes pour les études pathologiques. C'est pourquoi, un autre des objectifs de **NuQleoSim** est la mise au point d'une modélisation de l'activité du nucléole tout en proposant un suivi en temps réel de la simulation via une visualisation en 3D. À l'issue de cette simulation, des résultats sont proposés à l'utilisateur sous forme de graphiques ou de tableaux.

Table des matières

Introduction	1
1 Analyse	2
1.1 Contexte	2
1.2 Besoins fonctionnels	5
1.3 Besoins non fonctionnels	6
1.4 Etat de l'art	7
1.4.1 Bases de données relatives au nucléole	7
1.4.2 Les outils de modélisation	7
2 Conception	8
2.1 Base de données	8
2.2 Prototypage de l'interface	9
2.3 Modélisation du nucléole	12
2.3.1 Modélisation des composants du nucléole	12
2.3.2 Approche par système multi-agents	14
3 Réalisation	15
3.1 Langage et bibliothèques retenus	15
3.2 Diagramme de classe	17
3.3 Base de données	18

3.4	Modélisation du nucléole	20
3.4.1	Environnement	20
3.4.2	Agents	20
3.4.3	Déroulement	23
3.4.4	Résultats	23
3.5	Fonctionnement de l'interface	24
3.5.1	Onglet "Base de données"	25
3.5.2	Onglet "Simulation"	27
3.5.3	Onglet "Résultats"	30
4	Cas d'utilisation	33
4.1	Communication avec la base de données	33
4.2	Paramétrage et lancement d'une simulation	34
4.3	Présentation organisée des résultats	38
Conclusion		42
Appendices		45
A.	Diagramme du système multi agent	46
B.	Vue générale du diagramme de classe	48
C.	Diagramme de classe détaillé	50
Bibliographie		54

Introduction

La cellule est l'unité structurale constitutive des êtres vivants. L'existence ou non d'un noyau permet la distinction de deux types cellulaires : respectivement eucaryote et procaryote. La cellule eucaryote est subdivisée en structures spécialisées, appelées organites, auxquelles appartient le noyau. Sa fonction principale est le stockage de l'information génétique : l'ADN. Il est organisé en trois zones : l'euchromatine, l'hétérochromatine et le nucléole. Ce dernier, encore mal connu semble être impliqué dans de nombreuses maladies. C'est pourquoi il est aujourd'hui de plus en plus étudié.

Afin de mieux comprendre les processus internes au nucléole, il existe deux types d'approches. L'expérience en laboratoire malgré sa pertinence biologique coûte encore cher et prends du temps. Son alternative, la simulation *in silico*, réduit fortement les coûts à l'instar d'une pertinence biologique moindre. C'est pourquoi, les outils de modélisation du nucléole sont à développer dans le futur.

L'objectif de ce projet est de réaliser une application (**NuQleoSim**) regroupant la gestion d'une base de données relatives au nucléole ainsi qu'un outil permettant la simulation de l'activité du nucléole. Elle propose également un module d'analyse des résultats de la simulation.

Dans un premier temps, le contexte biologique permettra de préciser les attentes auxquelles répondra l'application en décrivant les besoins fonctionnels et non fonctionnels. Ensuite, la conception et l'implémentation qui en découle seront détaillées. Enfin, un cas d'utilisation de **NuQleoSim** sera présenté.

1. Analyse

1.1 Contexte

Généralités sur le nucléole

Domaine nucléaire dynamique, le nucléole est le centre de synthèse des ribosomes, à l'origine de la synthèse des protéines. Son état reflète le niveau d'activité de la cellule. En effet, suivant la phase du cycle cellulaire dans laquelle la cellule se trouve, le nucléole est plus ou moins actif. Il peut même venir à disparaître totalement lors de la mitose.

Biogenèse des ribosomes

Un ribosome comprend deux sous-unités, une petite et une grande. Ces sous-unités sont des macro-molécules constituées d'ARN et de protéines.

L'ADN présent dans le nucléole correspond aux gènes codant pour cet ARN ribosomique ou ARNr (voir la figure 1.1). Il y est transcrit en pré-ARNr par l'ARN polymérase I. Le pré-ARNr est ensuite scindé en 3 ARNr durant sa maturation.

Parallèlement, les gènes des protéines ribosomiques sont transcrits en ARN messagers par l'ARN polymérase II dans le noyau. Cet ARNm est alors exporté vers le cytoplasme où il sera traduit en protéines. Les protéines ribosomiques ainsi produites sont alors importées vers le nucléole où elles seront associées aux ARNr pour former les sous-unités ribosomiques.

Ces dernières sont enfin exportées vers le cytoplasme où elles pourront remplir leur fonction.

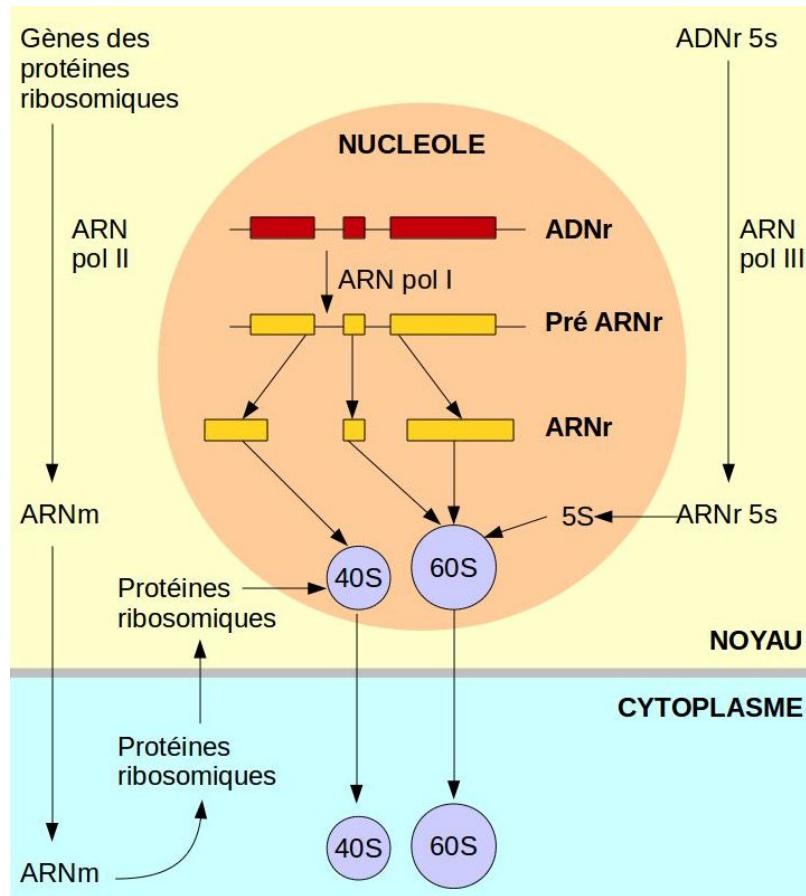


FIGURE 1.1 – Biogenèse des ribosomes

Morphologie du nucléole

Le nucléole ne possède pas de membrane. Cependant, chez les eucaryotes supérieurs, il est généralement composé de trois régions morphologiquement distinctes et visualisables en microscopie électronique :

- un ou plusieurs centre(s) fibrillaire(s),
- un centre fibrillaire dense (ou composant fibrillaire dense) autour du premier centre fibrillaire,
- et une région granulaire (ou composant granulaire) en périphérie.

Selon les organismes, les proportions des volumes de ces trois compartiments peut varier. Les séparations entre ces domaines ne sont pas nettes. Chez les plantes, des cavités (ou vacuoles) sont retrouvées en plus de ces territoires.

Le centre fibrillaire : il ne représente que 1 à 2 % du volume total du nucléole. Il contient de l'ARN polymérase I et des facteurs de liaison amont (ou UBF pour Upstream Binding Factor, en anglais).

Le centre fibrillaire dense : il représente environ 17 % de la fraction nucléolaire. Son volume reflète à peu près l'engagement nucléolaire dans la biogénèse des ribosomes.

La région granulaire : elle occupe environ 75 % du volume total du nucléole. La granulosité de cette région périphérique est due à la présence de particules ribonucléoprotéiques.

Maladies génétiques liées au nucléole

Des mutations au niveau de gènes codant pour des protéines nucléolaires sont associées à des maladies humaines. La variété des maladies associées telles que le syndrome de Werner, le syndrome de Treacher Collins ou encore la dyskératose congénitale, indique l'importance fondamentale du rôle du nucléole. La compréhension de sa dynamique ainsi que de sa structure est un premier pas vers le développement de nouvelles thérapies.

Les outils d'étude du nucléole

Le biologiste qui étudiait le nucléole avait pour habitude de répertorier l'ensemble des molécules étudiées dans un fichier de type tableur (Excel ou OpenOffice Calc). Par ailleurs, il n'existe pas encore de simulation *in silico* spécifique au nucléole.

NuQleoSim a donc pour objectif d'assister le biologiste dans son travail, en lui offrant une alternative à sa méthode de stockage de l'information, ainsi qu'un outil lui permettant de réaliser des simulations afin de mieux préparer les expériences en laboratoire.

1.2 Besoins fonctionnels

Stockage de données liées au nucléole

Les données qualitatives et quantitatives ainsi que les images de microscopie électronique récupérées sur les objets biologiques du nucléole doivent être conservées dans une base de données organisée et accessible. Les objets biologiques concernés sont les principaux constituants du nucléole soit des protéines, de l'ADN, des ARN, des ribosomes et l'ensemble des réactions qui se déroulent dans cette ultra-structure. Ces données devront être standardisées afin d'être aisément exportables dans d'autres bases de données.

Il doit être possible de saisir, de visualiser et de modifier les données contenues dans cette base. Les objets biologiques du nucléole doivent être décrits de manière organisée afin de faciliter la recherche et l'accès de ces objets. Un lien vers des bases de données comme SwissProt ou Protein Data Bank par exemple pourra être mis en place. Les images de microscopie électronique recueillies sur le nucléole devront être regroupées et gérées par cette banque de données.

Pour conserver les données accumulées et pouvoir les mettre à jour ou simplement les consulter, la banque de données doit être sauvegardée à chaque modification ou fermeture de l'application. Elle doit également être chargée à chaque lancement de l'application.

Simulation du nucléole

Dans un second temps l'application doit gérer une simulation sur le nucléole. Pour cela la saisie des paramètres doit être intuitive. L'utilisateur doit également pouvoir utiliser commodément les données contenues dans la banque de données décrite ci-avant. Une fois la simulation initiée celle-ci doit être visualisable en temps réel par l'utilisateur sans pour autant nuire à l'utilisation du reste de l'application. L'utilisateur doit également gérer la simulation lancée grâce à des fonctions de *pause*, *lecture* et *stop*. Une fois la simulation terminée les résultats s'affichent pour permettre une analyse. Les résultats doivent pouvoir être traités sous forme de tableaux ou de graphes personnalisables que l'utilisateur peut régénérer à tout instant. Les résultats doivent également pouvoir être exportés.

1.3 Besoins non fonctionnels

Ce projet devra également répondre à un certain nombre de besoins non fonctionnels permettant une utilisation intuitive. En particulier l'application doit être ergonomique et intuitive pour l'utilisateur.

Portabilité

Afin de faciliter la diffusion de **NuQleoSim**, il devra pouvoir fonctionner dans un maximum d'environnements d'exécution. Pour cela, il faudra utiliser pour créer cette application un langage et des bibliothèques disponibles pour GNU/Linux, Windows et MacOSX. Enfin, pour que la simulation puisse être affichée en temps réel et en 3D dans tous ces environnements, un outil de visualisation 3D multi-plateforme sera nécessaire.

Robustesse

L'utilisateur de **NuQleoSim** sera amené à saisir des données à deux niveaux. Le premier correspond à l'interface de saisie des informations de la base de données : il faudra contrôler le format des fichiers importés, ainsi que la nature des informations données pour certains champs (liens, concentrations etc.).

Le second porte sur le paramétrage de la simulation et de son analyse. Pour limiter les erreurs de saisie, la sélection de leurs paramètres se fera à l'aide de listes déroulantes contenant des données présentes dans la base.

Dans le cas où l'application rencontrerait un problème, lors du chargement d'un fichier ou lors de la simulation par exemple, l'utilisateur devra en être averti par un message lui indiquant le type d'erreur.

Modularité et extensibilité

L'application comprend deux parties : stockage de données et simulation. Elle sera donc organisée en modules. Cela permettra de répartir le travail de développement et éventuellement de réutiliser ou ajouter des portions de code dans l'avenir.

Performances

Lorsqu'une simulation est en cours, elle ne doit pas bloquer l'interface qui doit demeurer fluide. Par ailleurs, il pourrait être intéressant d'activer l'utilisation du GPU pour la simulation si celui-ci est détecté. Cette opération doit cependant rester transparente pour l'utilisateur.

1.4 Etat de l'art

1.4.1 Bases de données relatives au nucléole

Les bases de données spécialisées sur le nucléole sont peu répandues. Voici deux bases de données qui possèdent des informations sur les molécules du nucléole :

NOPdb: Nucleolar Online Proteomics Database [1] : <http://www.lamondlab.com/NOPdb3.0/>. NOPdb est une base de données du protéome humain du nucléole. Les données sont issues de la spectrométrie de masse et la base contient à ce jour plus de 50 000 peptides correspondant à environ 4500 protéines humaines du nucléole. L'API semble accessible sur demande.

LOCATE [4] : <http://locate.imb.uq.edu.au/>. LOCATE est une base de données axée sur la localisation subcellulaires des protéines de la souris et de l'Homme. Les localisations ont été déterminées par immunofluorescence à haut débit. Les protéines du nucléole sont au nombre de 1380 pour la souris et 1623 pour l'Homme. La base de données complète regroupe environ 60 000 protéines.

1.4.2 Les outils de modélisation

Il n'existe pas à ce jour d'outil capable de simuler l'activité du nucléole en prenant en compte toute sa dynamique et ses caractéristiques. On pourra cependant citer une application telle que NetLogo [3] qui est un environnement de modélisation orienté système multi-agent.

2. Conception

Dans ce chapitre, le fonctionnement de l'application sera décrit de manière détaillée, tout d'abord pour la partie base de données, puis pour l'interface et enfin pour la partie modélisation.

2.1 Base de données

La base de données doit contenir deux types très différents de données. Des expériences, principalement des photos de microscopie, ainsi que des molécules. La base de données a donc été conçue de manière à séparer ces deux types de données.

L'élément **Molecule** contient les données relatives aux molécules. Afin d'assurer une interopérabilité des informations, les identifiants de trois grandes bases de données biologiques ont été inclus (Protein Data Bank (PDB), EMBL et UniprotKB). D'autres liens vers des bases de données peuvent être ajoutés dans le cas d'une poursuite du développement de **NuQleoSim**.

L'élément **Experiment** contient, lui, les données relatives à des expériences. Les données sont principalement une liste d'images ainsi que des données courantes telles que l'auteur et la date.

Un champ supplémentaire, **comments** est présent pour chacune des deux classes d'éléments et permet à l'utilisateur de **NuQleoSim** de laisser des informations complémentaires de différents types.

Afin de répondre à toutes ces caractéristiques, la base de données de **NuQleoSim** est organisée selon le schéma qui suit :

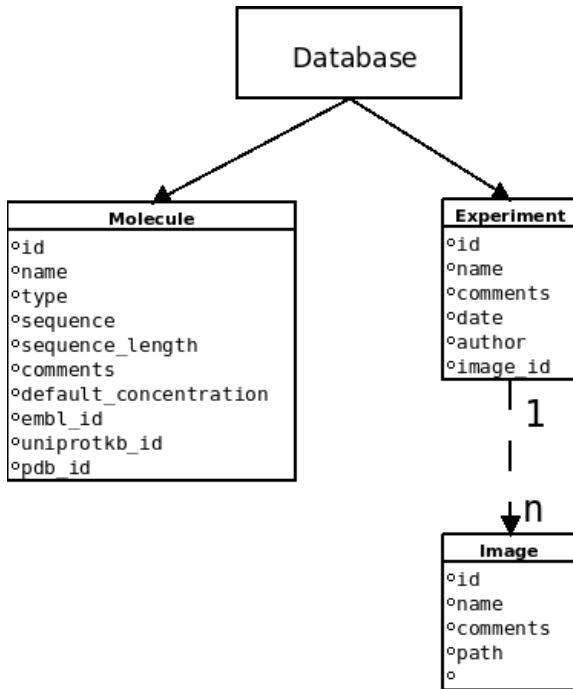


FIGURE 2.1 – Schéma de la base de données.

2.2 Prototypage de l'interface

NuQleoSim a deux rôles principaux. Le premier est de permettre la communication avec la base de données. Le second, ayant trait à la simulation, est lui-même subdivisé en deux : la simulation en elle-même ainsi que l'affichage des résultats de la simulation.

L'interface est donc divisée en fonction de ces trois fonctions principales :

- Le premier onglet (Fig. 2.2) est l'interface de communication avec la base de données : l'utilisateur peut y ajouter, modifier ou supprimer des données dans la base.
- Le second onglet (Fig. 2.3) permet le paramétrage de la simulation et son démarrage.
- Le troisième onglet (Fig. 2.4) ne s'active qu'en présence de résultats. Dans ce cas, il permet la génération des résultats sous différentes formes (graphique et tableau).

La visualisation de la simulation est une tâche lourde qui pourrait, par la suite, demander à être désactivée à la demande. Lors du démarrage de la simulation, la visualisation se fait donc dans une fenêtre séparée de la fenêtre principale.

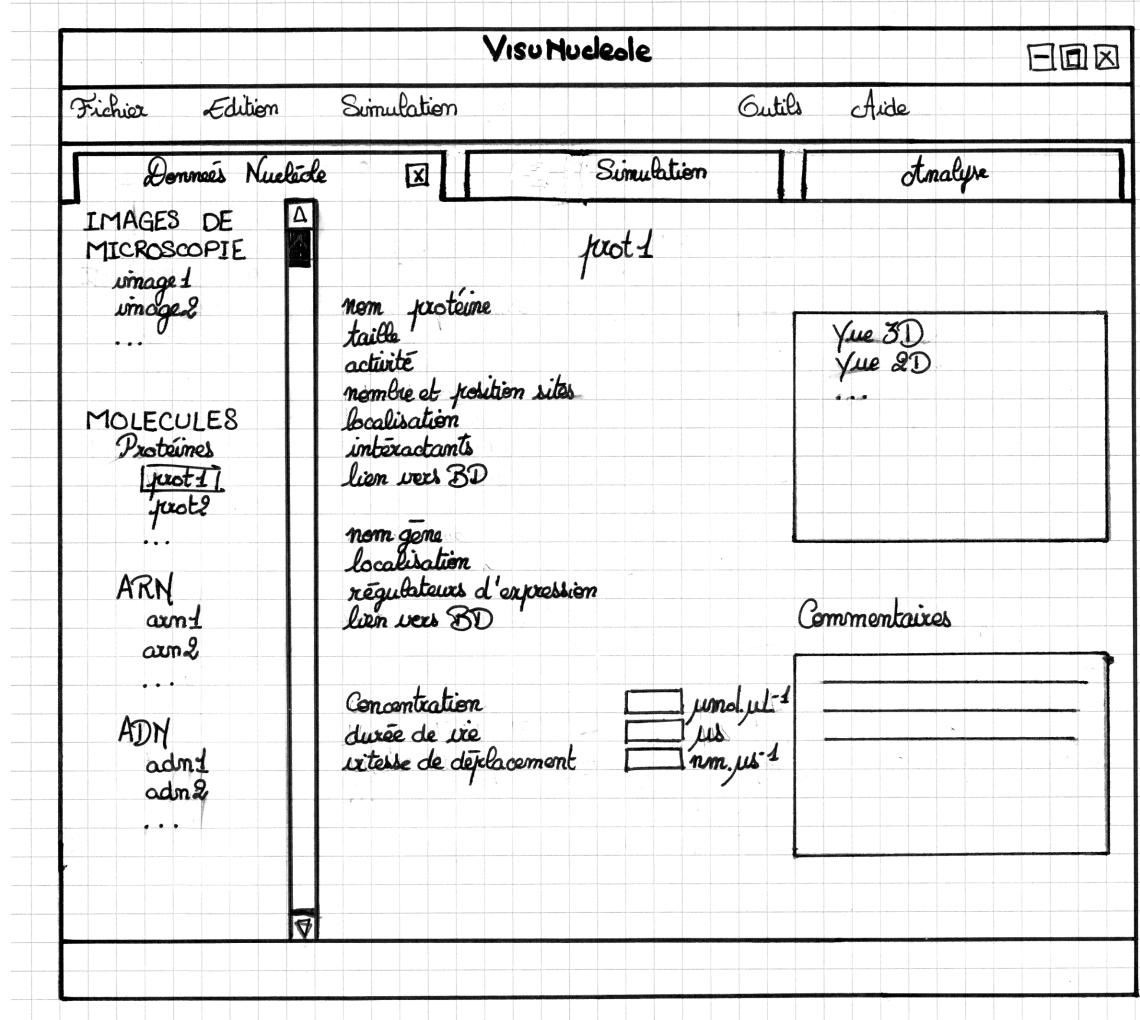


FIGURE 2.2 – Premier onglet : interface avec la base de donnée.

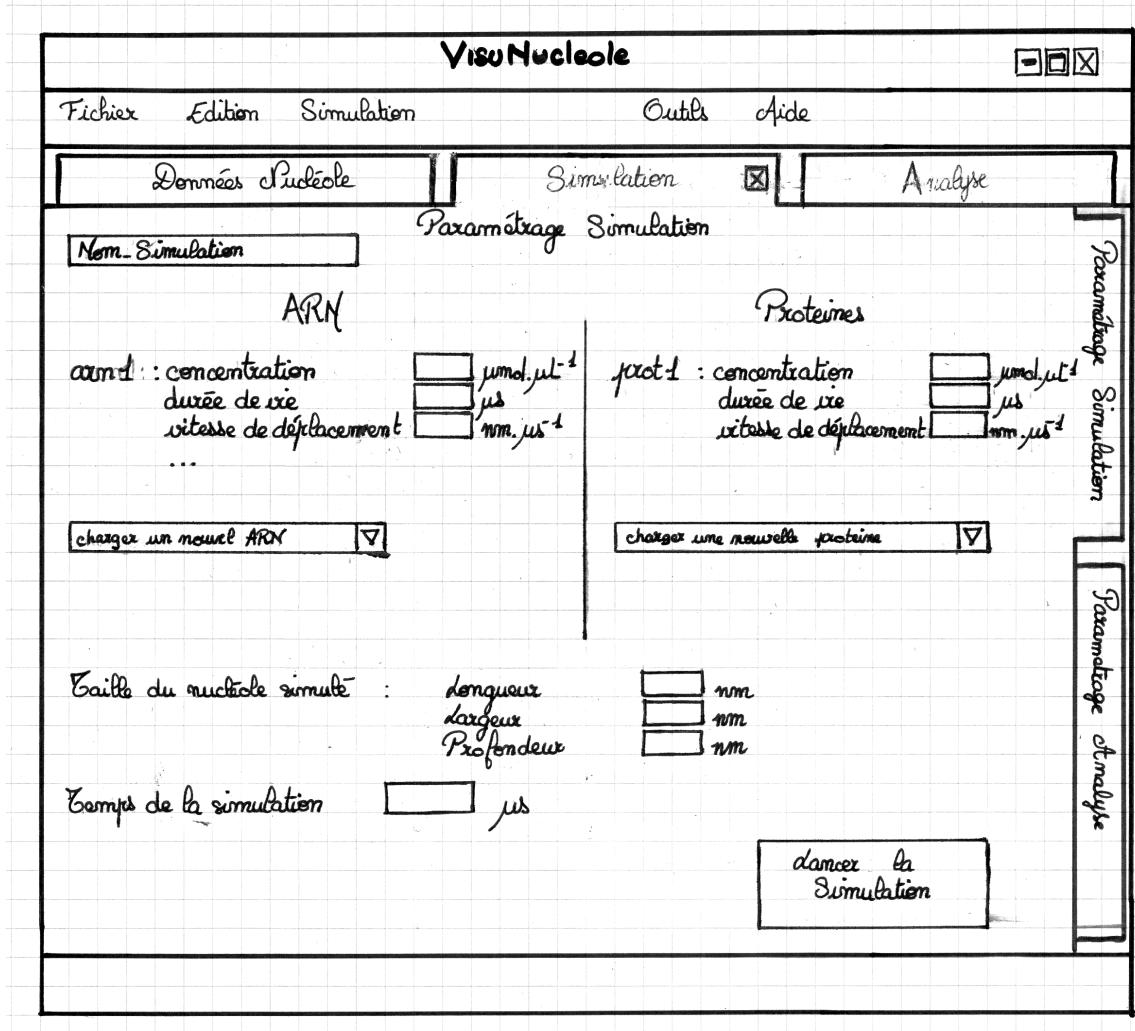


FIGURE 2.3 – Second onglet : paramétrage et lancement de la simulation.

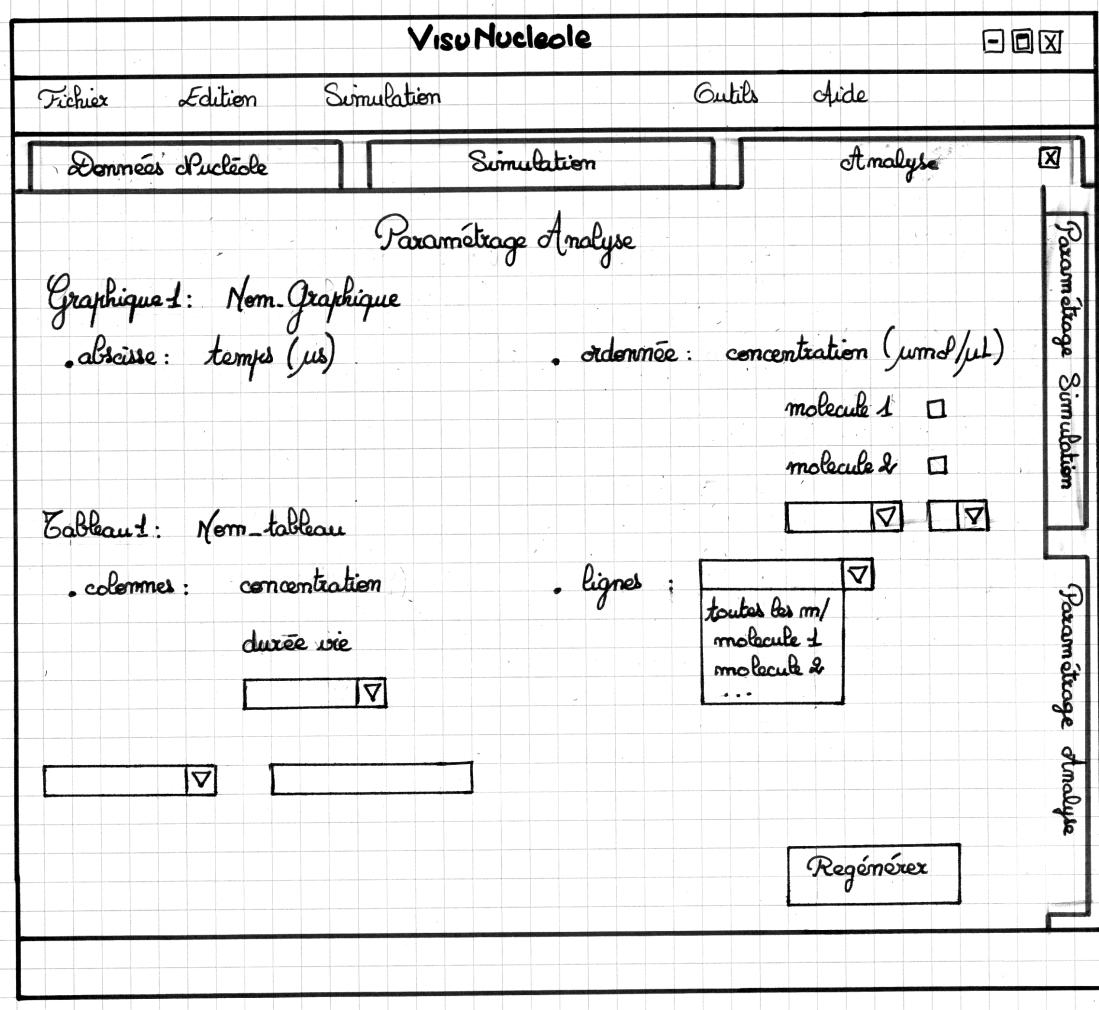


FIGURE 2.4 – Troisième onglet : génération et mise en forme des résultats.

2.3 Modélisation du nucléole

2.3.1 Modélisation des composants du nucléole

Le nucléole est une structure du noyau des cellules eucaryotes, mais ce n'est pas un compartiment. En effet, aucune paroi ne le délimite, seule sa composition permet de l'identifier. Comme il a été vu précédemment, il est le centre de la production intense des ribosomes. Sa densité est donc supérieure à celle du reste du noyau et

son apparence en microscopie est, par conséquent, elle aussi différente ([2] et [5]).

La localisation dans le noyau du nucléole n'ayant aucun impact important pour la réalisation de ce projet, la modélisation du nucléole commence donc par la modélisation de ce composant lui-même. En outre, il a été observé que le nucléole lui-même comptait plusieurs composants bien distincts. Il comprend le DFC (*Dense Fibrillar Component*), le FC (*Fibrillar Component*) et GC (*Granular Center*). Cependant ce sont des composants et non des compartiments, c'est donc leur composition qui les caractérise et non la présence d'une membrane délimitant les divers compartiments. Le nucléole modélisé doit prendre en compte cette spécificité. Il doit donc comprendre trois zones distinctes représentant respectivement les trois composants.

Comme cela vient d'être vu, chaque composant du nucléole se distingue de par sa composition en protéines et en ARN. Cette caractéristique doit donc, elle aussi, être prise en compte. C'est-à-dire qu'il doit être possible de préciser, pour chaque molécule ayant un rôle dans la simulation, les composants dans lesquels elle se déplace. Enfin, c'est au coeur du nucléole que se déroule la transcription des ARN ribosomiaux. Il y a donc synthèse puis maturation de ces ARN dans les différents composants du nucléole. Ceci constitue le dernier point à modéliser pour mettre en place une simulation. En effet, les ARN sont transcrits au niveau du FC. Puis, ils subissent plusieurs maturations en traversant le DFC puis le GC, pour sortir du nucléole et du noyau et être assemblés en ribosomes. La modélisation du nucléole doit donc également prendre en compte cette autre caractéristique.

En résumé, la modélisation du nucléole doit prendre en compte les caractéristiques de l'environnement « nucléole » :

- zone délimitant la composante FC
- zone délimitant la composante DFC
- zone délimitant la composante GC
- position des molécules dans l'environnement : deux molécules ne peuvent pas se superposer.

Cette modélisation doit également prendre en compte les molécules intervenant dans la simulation :

- protéine : sa position, sa vitesse de déplacement, les composantes dans lesquelles elle prend part.

- ARN : sa position, sa vitesse de déplacement, les composantes dans lesquelles elle prend part et son état de maturation, sa vitesse de maturation.
- « transcriptor » : sa position dans le FC, sa vitesse de déplacement, sa vitesse de « transcription ».

2.3.2 Approche par système multi-agents

Un système multi-agents (SMA) est un système composé d'un ensemble d'agents, situés dans un certain environnement et interagissant selon certaines relations. Un agent est une entité caractérisée par le fait qu'elle est, au moins partiellement, autonome.

Cette approche est de plus en plus utilisée pour modéliser des systèmes complexes. L'avantage principal de ce type de modélisation est sa simplicité de mise en place pour le modélisateur. Il suffit de décrire le comportement de chaque agent en fonction de son environnement. Ce sont souvent des rôles simples et issus de l'observation (dans le cas des modélisations en biologie). Cependant ce type d'approche autorise une certaine complexification aussi bien du comportement des agents que des interactions entre agents et avec leur environnement.

Ainsi en décrivant le nucléole ainsi que le comportement des molécules qu'il contient, il est possible d'appliquer une approche multi-agents (Annexe A.).

3. Réalisation

3.1 Langage et bibliothèques retenus

La nécessité d'intégrer une simulation et une visualisation 3D en temps réel a conduit à développé **NuQleoSim** en C++. Cependant, la complexité de développement qu'implique parfois le C++ est largement compensée par l'utilisation du framework Qt. En effet, dans ce projet Qt est utilisé en tant que framework et non en tant qu'une simple bibliothèque GUI comme GTK ou wxWidget. L'architecture de **NuQleoSim** repose donc principalement sur Qt, cependant certain module comme celui de la modélisation reste indépendant de Qt afin de le rendre facilement portable en supprimant la dépendance à Qt.

Qt implique des temps de compilation relativement long et des executables souvent de grosses taille. Cependant, ceci est largement compensé par plusieurs avantages tel que la rapidité d'exécution des applications Qt, la rapidité de développement grâce à une API très bien documenté et la présence d'une large communauté. De plus, Qt est totalement multiplateforme.

Voici une liste des technologies utilisées lors de la réalisation de **NuQleoSim**.

Framework Qt (<http://qt.nokia.com/products/>) : est un framework orienté objet et développé en C++ par Qt Development Frameworks, filiale de Nokia. Elle offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des files d'exécution, d'analyse XML, etc. En plus d'être très complet comme framework, un autre avantage est son aspect multi plateforme. En effet, par simple recompilation du code source, un programme Qt pourra être porté vers des OS tel que GNU/Linux, Windows et Mac OS X.

libQGLViewer (<http://www.libqglviewer.com/>) : est une bibliothèque basé sur Qt qui facilite la création de scène OpenGL. Il offre certaines fonctionnalités typiques des visionneurs 3D, comme la possibilité de déplacer la caméra en utilisant la souris, ce qui manque à la plupart des autres API. Les autres caractéristiques comprennent la manipulation de la souris, la superposition de widget vectorisé aux scènes 3D, la sélection d'objets. Il permet de créer des applications 3D complexes rapidement. Cette bibliothèque est utilisé pour la visualisation en temps réel lors de la simulation.

Qwt (<http://qwt.sourceforge.net/>) : est une bibliothèque basé sur Qt qui fournit un ensemble de widget destiné à faciliter l'affichage de graphique destiné aux applications techniques et scientifiques. Elle est utilisé pour l'affichage des résultats issus de la modélisation.

XML (<http://www.w3.org/XML/>) : est un langage informatique de balisage générique. Il sert essentiellement à stocker/transférer des données de type texte Unicode, structurées en champs arborescents. Il sera utilisé pour le stockage de la base de données d'objets biologiques. En effet, cette solution contrairement à des SGBD plus classiques tel que MySQL a l'avantage d'être indépendant (pas besoin de serveur SQL) et de faciliter l'échange de base de données (simple fichier XML). La cohérence de la base de données sera assurée par un fichier XML Schema qui est un langage de description de format de document XML permettant de définir la structure d'un document XML.

CMake (<http://www.cmake.org/>) : est est un « moteur de production » multiplate-forme. Il permettra de recompiler facilement l'ensemble du code source sous différents OS. Il possède aussi des outils de génération de fichiers d'installation (.deb, .rpm, .exe, .tar.gz).

Git (<http://git-scm.com/>) : est est un logiciel de gestion de versions décentralisée. Son choix a été motivé par le fait qu'il soit open-source d'une part et d'une autre part pour sa grande flexibilité. Git est capable de gérer des cas complexes de conflits de fusion entre plusieurs versions. De plus, Git, créé par Linus Torvald, est

aujourd’hui utilisé pour la gestion des sources du kernel de Linux. On imagine donc que si cet outil convient à la gestion d’un aussi gros projet (11,5 millions de lignes de codes) avec autant de collaborateurs, il conviendra parfaitement à un projet plus modeste.

3.2 Diagramme de classe

L’architecture est composé de plusieurs modules qui communiquent entre eux principalement via le mécanisme de signal et slot de Qt (un tel mécanisme existe aussi dans d’autre bibliothèque connu tel que Boost).

Une version simplifié et une version détaillée sont présentes en annexe de ce rapport (B. et C.). Voici un bref description des principaux modules :

- Le modèle de donnée est un ensemble de classe destiné à modéliser les données manipulées lors de l’interfaçage avec la base de donnée.
- **BioParser** est une classe abstraite qui permet de facilement écrire des classes capables de parser n’importe quel format de fichier biologique. Pour le moment, **NuQleoSim** est capable de parser les fichiers de type PDB.
- Le module **DataUI** et ses classes associés sont un ensemble de classe widget chargé de l’interfaçage avec la base de donnée.
- Le module de base de donnée permet la communication du reste de l’application avec n’importe quel type de base de donnée via une classe abstraite **DBAccess**. A ce jour, **NuQleoSim** repose sur une base de donnée de type XML.
- Le module **SimuUI** et ses classes associés sont un ensemble de widget chargé du paramétrage de la simulation. Son rôle est aussi de gérer le lancement et le suivi de la modélisation ainsi que la mise à jour de la visualisation.
- **Computing Module** est chargé de la simulation du nucléole. Il ne dépend pas de Qt afin de le rendre facilement portable. Le système d’héritage permet, dans l’avenir, l’ajout de manière de modéliser supplémentaires. Pour le moment, les calculs sont effectués sur le CPU, mais il a été prévu de permettre l’ajout de classe effectuant les calculs sur le GPU via différentes librairies tel que OpenCL ou encore CUDA.
- **3D Module** est chargé d’afficher l’espace cellulaire représentant le nucléole lors de la simulation.

- **AnaUI** est un ensemble de widget capable de générer des résultats provenant d'une modélisation précédente.

3.3 Base de données

Dans un premier temps **NuQleoSim** doit permettre d'enregistrer des données relatives à des molécules ainsi qu'à des expériences. Une base de données a donc été mise en place. Elle permet dans un premier temps la saisie par l'utilisateur des différents champs d'intérêt, que ce soit pour une molécule ou une expérience (Fig. 2.1). Dans le cas de la partie de la base de données gérant les molécules, l'utilisateur peut spécifier :

- **id** : l'identifiant dans la base de données de **NuQleoSim**
- **name** : le nom de la molécule
- **type** : le type moléculaire : ADN, ARN, protéine, ribosome, molécule
- **sequence** : la séquence de la protéine (quelque soit son format)
- **length** : la longueur de la séquence
- **comments** : les éventuels commentaires de l'utilisateur
- **concentration** : la concentration de la molécule dans le nucéole
- **embl_id** : l'identifiant de la molécule dans l'EMBL avec un lien vers la fiche
- **uniprot_id** : l'identifiant de la molécule dans UNIPROT avec un lien vers la fiche
- **pdb_id** : l'identifiant de la molécule dans la PDB avec un lien vers la fiche

Ainsi, il est possible de faire des modifications à tout instant. De plus, des liens vers la page web de la molécule dans diverses bases de données biologiques ont été mis en place pour compléter les caractéristiques des molécules. Afin de faciliter l'import de molécule, il a été mis en place un système de téléchargement automatique de fichier PDB via l'id PDB.

Concernant les expériences, l'implémentation est similaire, si ce n'est que les noms des champs à compléter est différent :

- **id** : l'identifiant dans la base de données de **NuQleoSim**
- **name** : le nom de la molécule
- **type** : le type moléculaire : ADN, ARN, protéine, ribosome, molécule

- **comments** : les éventuels commentaires de l’utilisateur
- **date** : l’identifiant de la molécule dans UNIPROT avec un lien vers la fiche
- **author** : l’identifiant de la molécule dans la PDB avec un lien vers la fiche

Cette base de données doit être rechargée à chaque ouverture de **NuQleoSim** et enregistrée à chaque modification. Voici, un aperçu d’un fichier XML de la base de donnée de **NuQleoSim** vide.

```
<nucleolus_db>
    <experiments>
        <experiment>
            <author/>
            <comments/>
            <date/>
            <name/>
            <images>
                <image>
                    <comments/>
                    <id/>
                    <name/>
                    <path/>
                <image>
            <images>
        </experiments>
        <molecules>
            <molecule>
                <coments/>
                <default_concentration/>
                <name/>
                <sequence/>
                <sequence_length/>
                <type/>
                <embl_id/>
                <pdb_id/>
                <uniprotkb_id/>
            <molecule>
        <molecules>
    
```

```
</nucleolus_db>
```

Enfin, il est possible de compléter les **expériences** par une ou des images. En effet, tout comme la saisie et la consultation des informations relative à une expérience ou une molécule, l'ajout d'images est pris en charge par la base de données.

3.4 Modélisation du nucléole

Comme vu lors de la conception, la modélisation du nucléole est basé sur une approche multi-agents. Les différentes molécules ou systèmes qui entrent en jeu peuvent être assimilés à des agents qui ont des caractéristiques, qui réalisent des actions et qui interagissent entre eux ou avec leur environnement.

3.4.1 Environnement

Afin de modéliser le nucléole où se déroule la simulation, un environnement tridimensionnel a été mis en place. Le nucléole est modélisé comme étant un parallélépipède rectangle dont les trois dimensions sont renseignées par l'utilisateur. En outre, les trois composantes du nucléole que constituent le FC, le DFC et le GC sont elles aussi assimilées à des structures parallélépipédiques tridimensionnelles dont les dimensions sont calculées via un pourcentage de l'environnement nucléolaire total. On retrouve ainsi au centre le FC entouré du DFC lui même entouré du GC (Fig. 3.1). Même si aucune membrane physique ne délimite ces différentes composantes dans la « réalité », afin de mieux les distinguer au cours de la simulation, il est possible de faire apparaître chaque compartiment dans une couleur distincte des deux autres.

3.4.2 Agents

Une fois l'environnement mis en place, il faut placer les différents éléments qui vont entrer en jeux dans la simulation. De plus, chaque élément possède des contraintes de par son type même ou imposées par l'utilisateur. Ainsi, les différents agents sont placés aléatoirement tout en respectant certaines règles :

- deux éléments ne peuvent pas être en même temps sur une même « case ».

- un élément ne peut pas être placé dans une composante dans laquelle il ne prend pas part.

Un nombre déterminé de chaque agent doit être placé pour initialiser la simulation. Pour cela l'utilisateur précise pour chaque molécule les composantes dans lesquelles il veut les situer et dans quelles proportions. Ainsi, une molécule peut se déplacer dans les trois compartiments, mais n'être présente au début de la simulation que dans l'un des trois par exemple (Annexe A.). Cependant si une molécule ne participe pas à la constitution d'une composante, il ne peut pas y avoir de molécules de ce type dans cette composante.

Au cours de la simulation, les agents peuvent se déplacer au sein de l'environnement. Dans le but de ne favoriser aucun agent devant un autre, ils sont déplacés chacun leur tour dans un ordre aléatoire. De plus, chaque molécule possède sa propre probabilité de déplacement. Ainsi, une fois un agent sélectionné aléatoirement, il est déplacé dans une case voisine, respectant les contraintes appliquées à l'agent, aléatoirement.

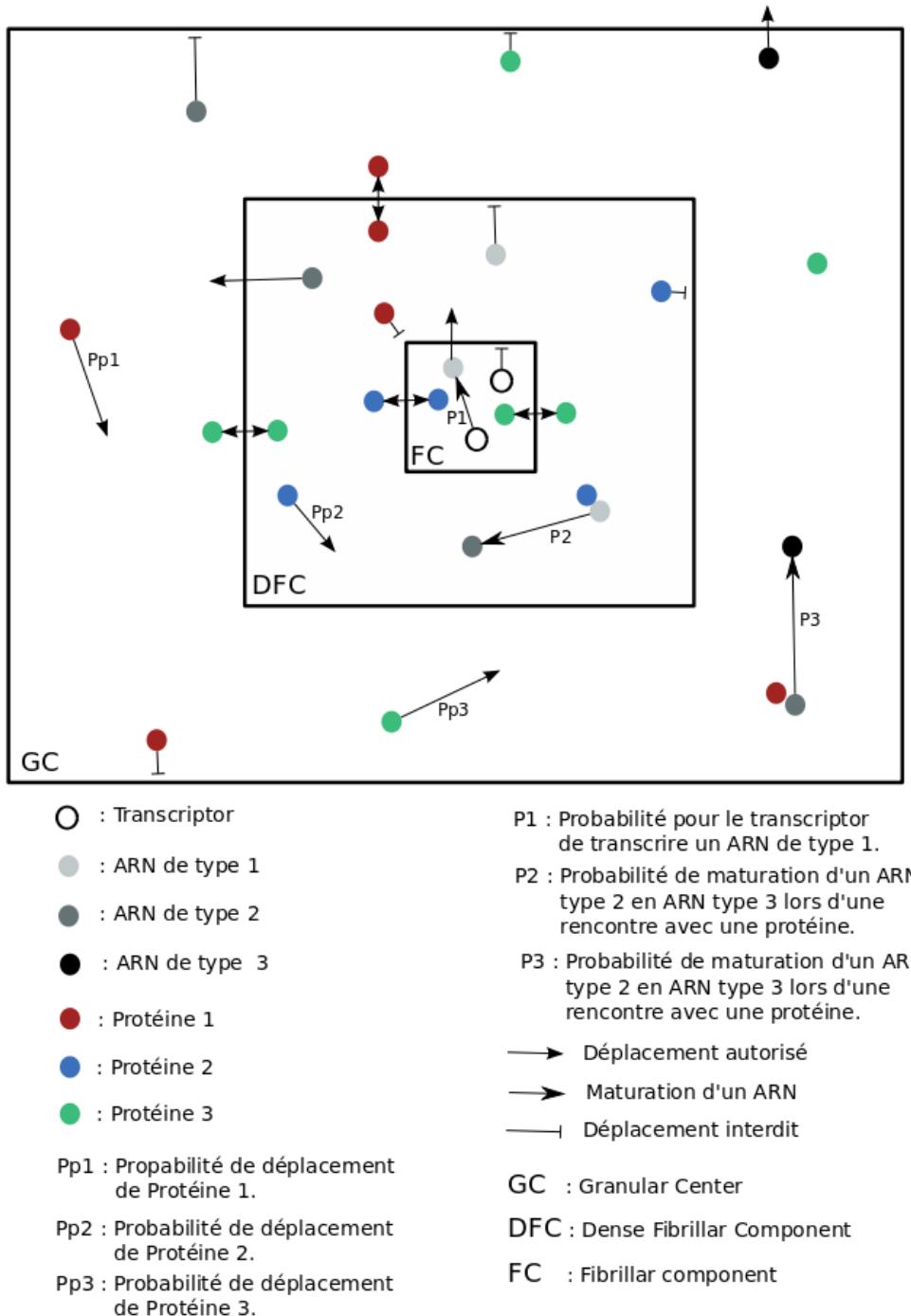


FIGURE 3.1 – *Interactions des différents agents entre eux et avec leur environnement.*

Lors des déplacements on distingue les molécules protéiques des molécules d'ARN. En effet, les ARN subissent certaines étapes de maturation avant de sortir du nucléole et du noyau. Pour modéliser ce comportement une probabilité de maturation d'un ARN à proximité d'une protéine a été définie. L'ARN maturé se distingue de sa forme précoce de par sa capacité à accéder à la composante nucléolaire suivante ou sa sortie du nucléole pour aller constituer les ribosomes.

3.4.3 Déroulement

Lorsque une simulation est initiée, les données sont enregistrées à chaque pas de temps. Soit : pour chaque type de molécules, leur quantité dans chaque composante est conservée. En parallèle, leurs mouvements sont calculés pour chaque pas de temps et transmis au module gérant la visualisation pour permettre l'affichage en trois dimensions de la simulation en cours (Fig. 3.6). Cet échange mis en place permet également de mettre en pause une simulation en cours ou de la stopper. Une fois le temps de simulation, spécifié par l'utilisateur, terminé l'ensemble des données rassemblées peuvent être traitées.

3.4.4 Résultats

Dans un dernier temps, une fois la simulation terminée, l'utilisateur peut sélectionner parmi les diverses molécules présentes dans la simulation, celles dont il veut observer l'évolution. Il peut alors décider d'afficher les résultats sous forme de graphe ou bien dans un tableur qu'il pourra exporter respectivement au format PNG ou CSV (Fig. 3.10 et Fig. 3.9).

3.5 Fonctionnement de l'interface

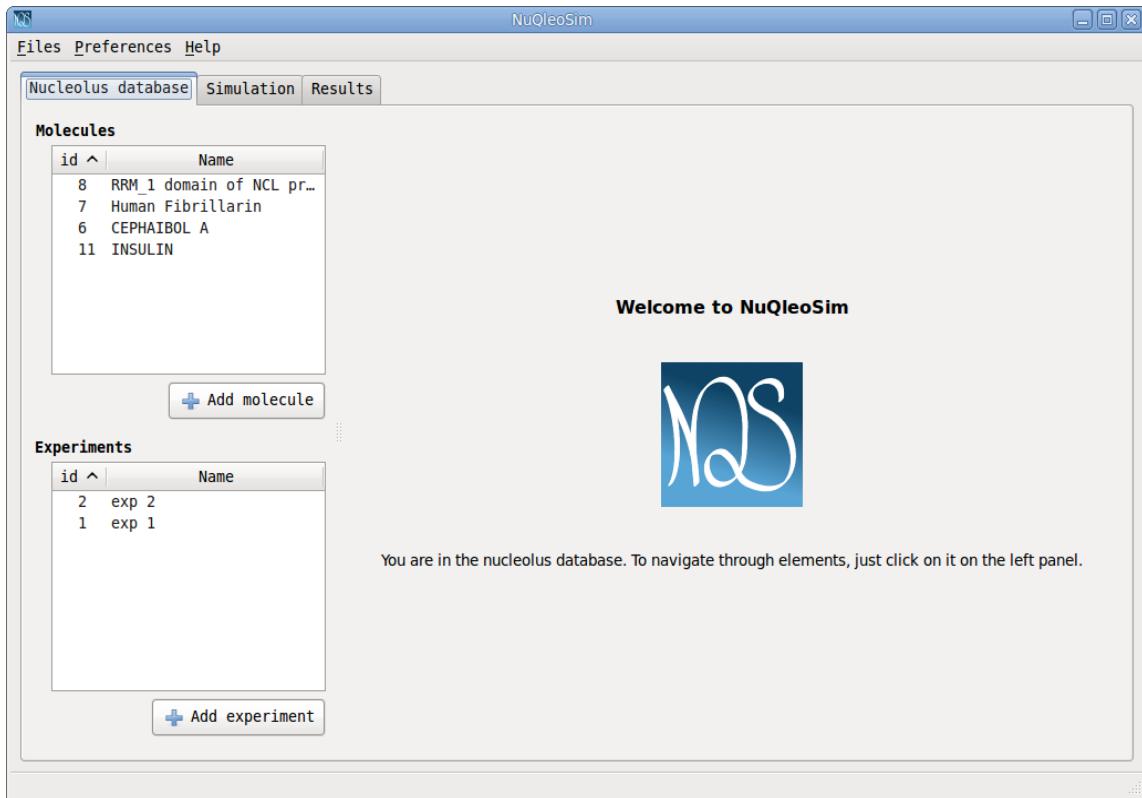


FIGURE 3.2 – Interface d'ouverture de *NuQleoSim*.

Tout comme l'implémentation du projet peut être scindée en trois parties principales, l'interface permettant à l'utilisateur d'utiliser l'application comprend à son ouverture trois parties principales, chacune associée à un onglet (Fig. 3.2).

3.5.1 Onglet "Base de données"

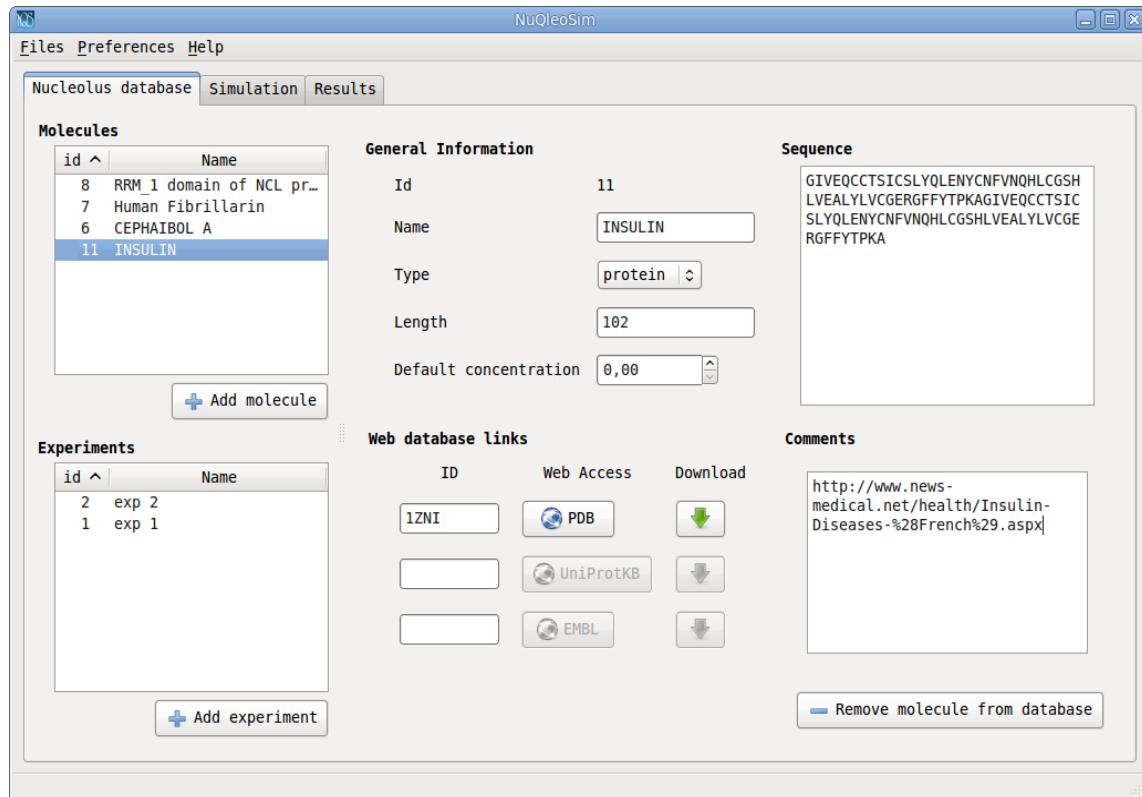


FIGURE 3.3 – Interface de saisie ou de modification de données sur une molécule.

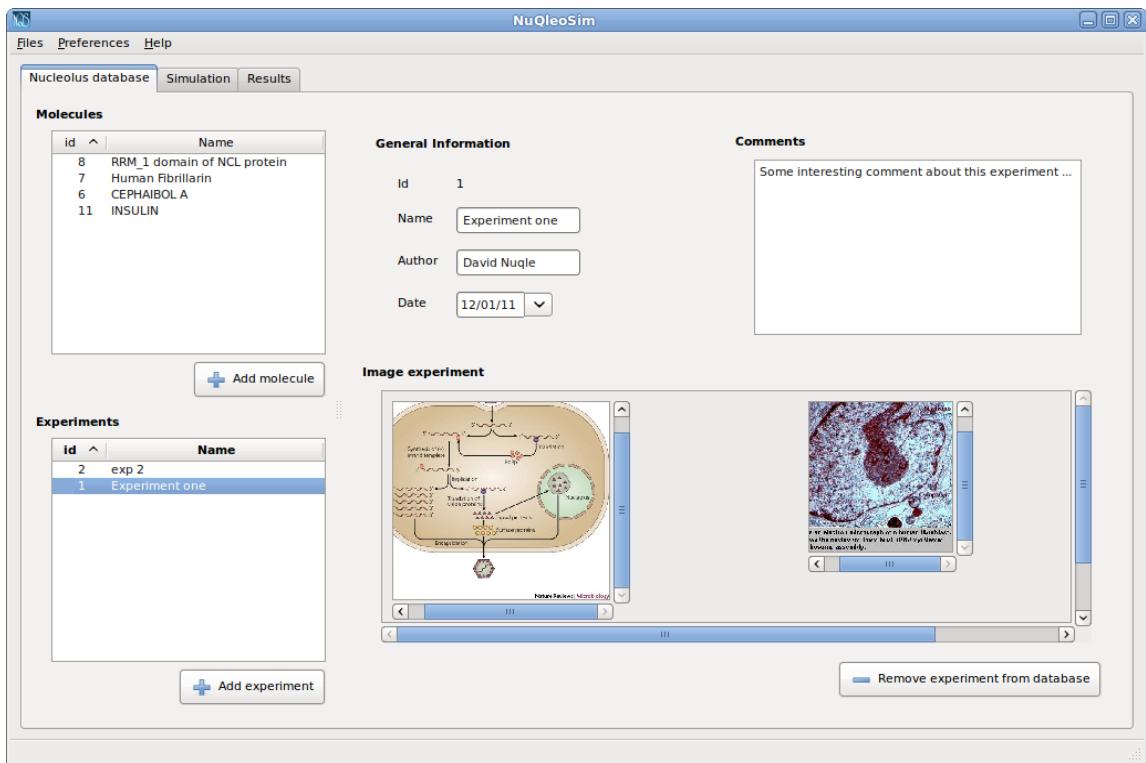


FIGURE 3.4 – Interface de saisie ou de modification des données d'une expérience.

Le premier onglet de **NuQleoSIm** permet d'accéder à la base de données constituée par les utilisateurs. L'interface permet la distinction entre les molécules pouvant prendre part à une simulation ou une expérience (Fig. 3.3), des expériences elles-mêmes (Figure ref BQExp). Pour chaque molécule, il est possible de rentrer son identifiant, son nom, sa taille, sa concentration, son identifiant dans diverses bases de données ainsi que sa séquence et de préciser sa nature dans une liste de molécules pré-rentrée (ADN, ARN, protéique, etc). Un lien est alors mis en place pour permettre d'accéder directement aux informations sur les bases de données internet spécifiées. Concernant l'enregistrement d'une expérience, il est possible, de même, de préciser divers champs comme l'identifiant de l'expérience, l'expérimentateur et la date. Dans le cas des expériences, des images peuvent être associées. Enfin, pour les deux un champs « comments » a été intégré pour permettre à l'utilisateur de rentrer des commentaires ou ajouter des données dont les champs n'existent pas encore.

3.5.2 Onglet "Simulation"

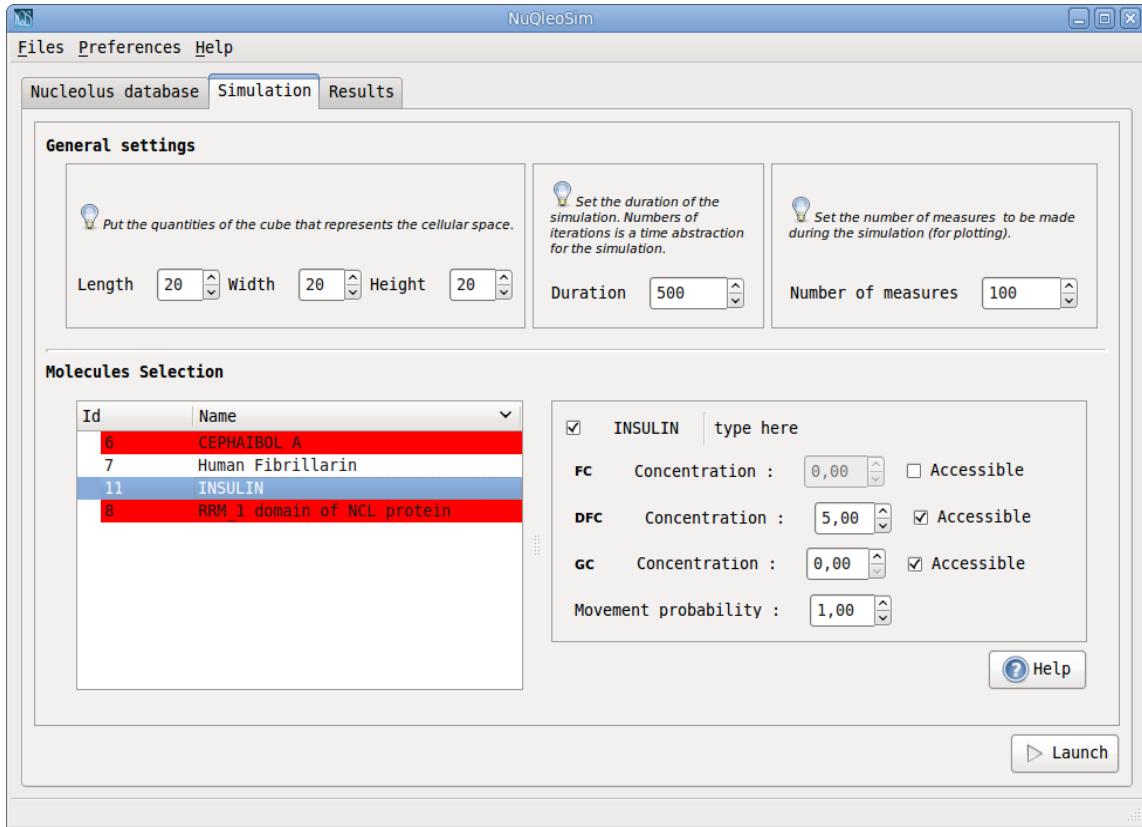


FIGURE 3.5 – Interface de paramétrage d'une simulation.

Le deuxième onglet de **NuQleoSim** permet le paramétrage d'une simulation (Fig. 3.5). Ce paramétrage peut être scindé en deux en distinguant les paramètres se rapportant au nucléole, des données se rattachant aux molécules intervenant dans la simulation. Une séparation spatiale sur l'interface est alors opérée. En haut les grandeurs spécifiant le nucléole modélisé et en dessous les variables discriminant les molécules. Les données relatives aux molécules sont accessibles en sélectionnant une molécule parmi celles de la base de données de **NuQleoSim**. L'utilisateur doit alors sélectionner (en cochant la case dédiée) celles qui l'intéressent, puis spécifier leurs différentes concentrations dans chaque composante ainsi que les composantes auxquelles elles peuvent avoir accès. Toute molécule sélectionnée est alors surlignée en rouge dans la liste, pour permettre à l'utilisateur d'avoir une vue d'ensemble de

sa simulation.

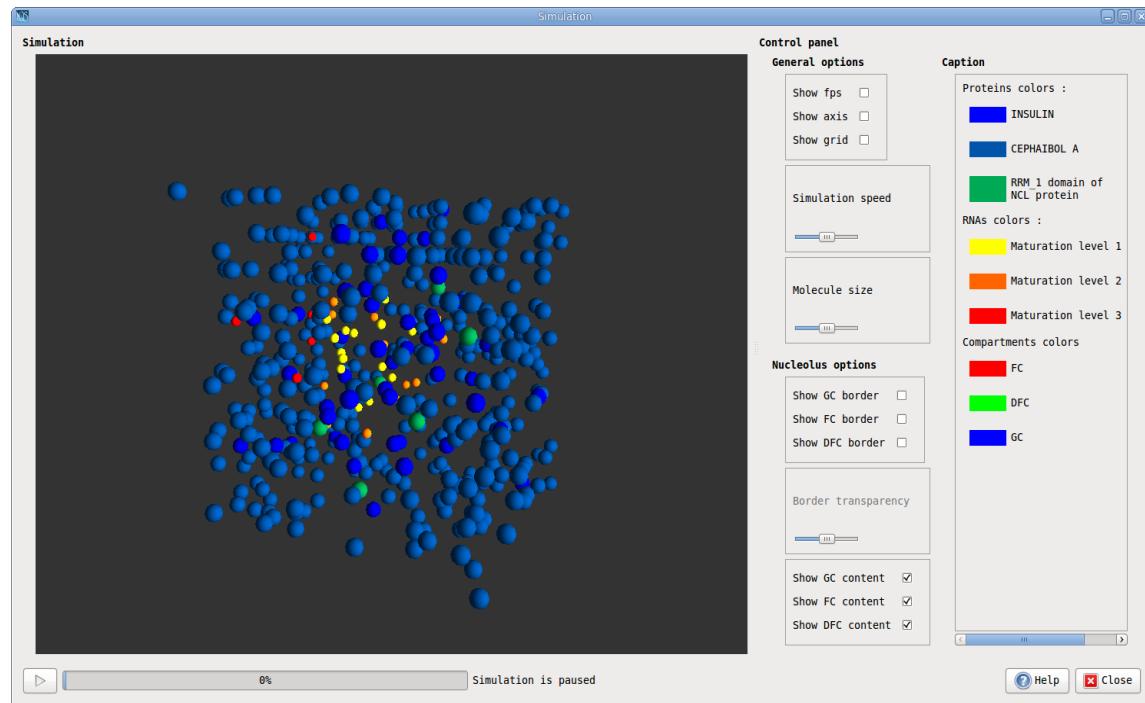


FIGURE 3.6 – Fenêtre de suivi en temps réel de la simulation.

Une fois la simulation paramétrée, l'utilisateur peut l'initier en cliquant sur le bouton « Launch ». Une nouvelle fenêtre s'ouvre alors (Fig. 3.6). La simulation en cours peut alors être mise en pause ou stoppée à tout instant. Afin de différencier aisément les différentes molécules en jeu, une couleur différente leur est attribuée dans la modélisation (une légende permettant de faire le lien). De plus dans l'optique de distinguer les ARN des autres molécules et pour rester plus ou moins en accord avec la réalité, les sphères modélisant les ARN sont plus petites. Enfin pour laisser une certaine marge de liberté à l'utilisateur, ce dernier peut moduler certains aspects de la simulation. Effectivement, il peut modifier la taille des molécules et la vitesse de la simulation. Il peut faire apparaître indépendamment les différentes composantes (Fig. 3.7) ou encore ne visualiser que les molécules présentes dans une ou plusieurs composantes. Pour terminer, des fonctions de zoom, de contrôle de la transparence des composantes, d'affichage des axes ou encore d'une grille sont disponibles.

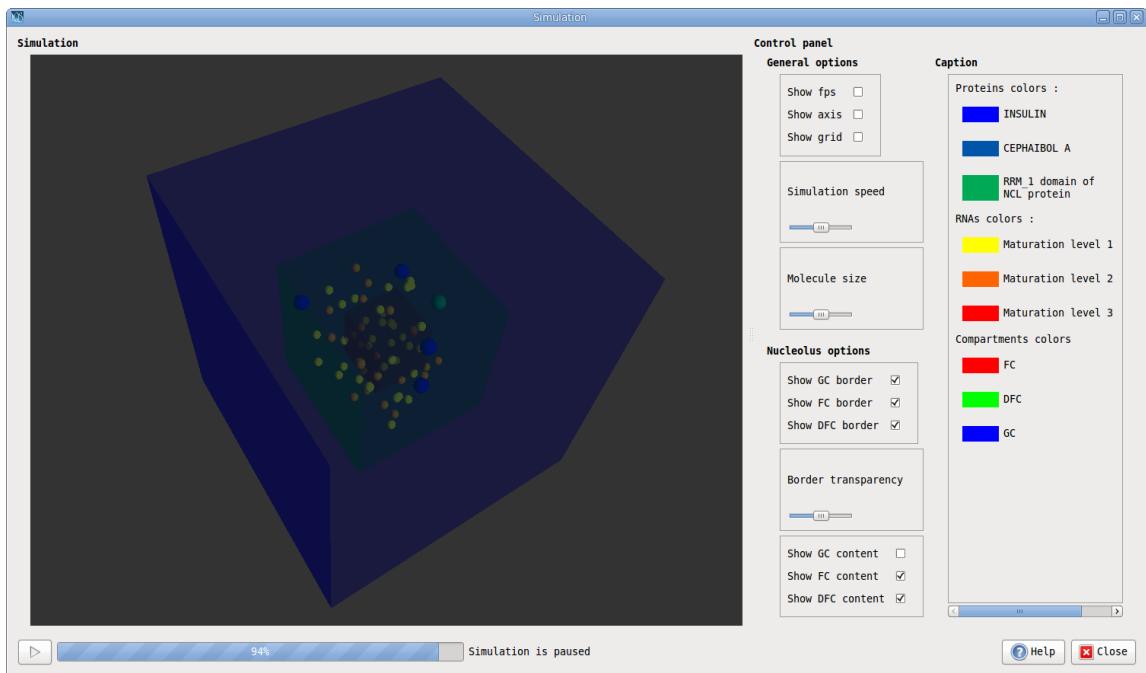


FIGURE 3.7 – Visualisation possible des différentes composantes.

3.5.3 Onglet "Résultats"

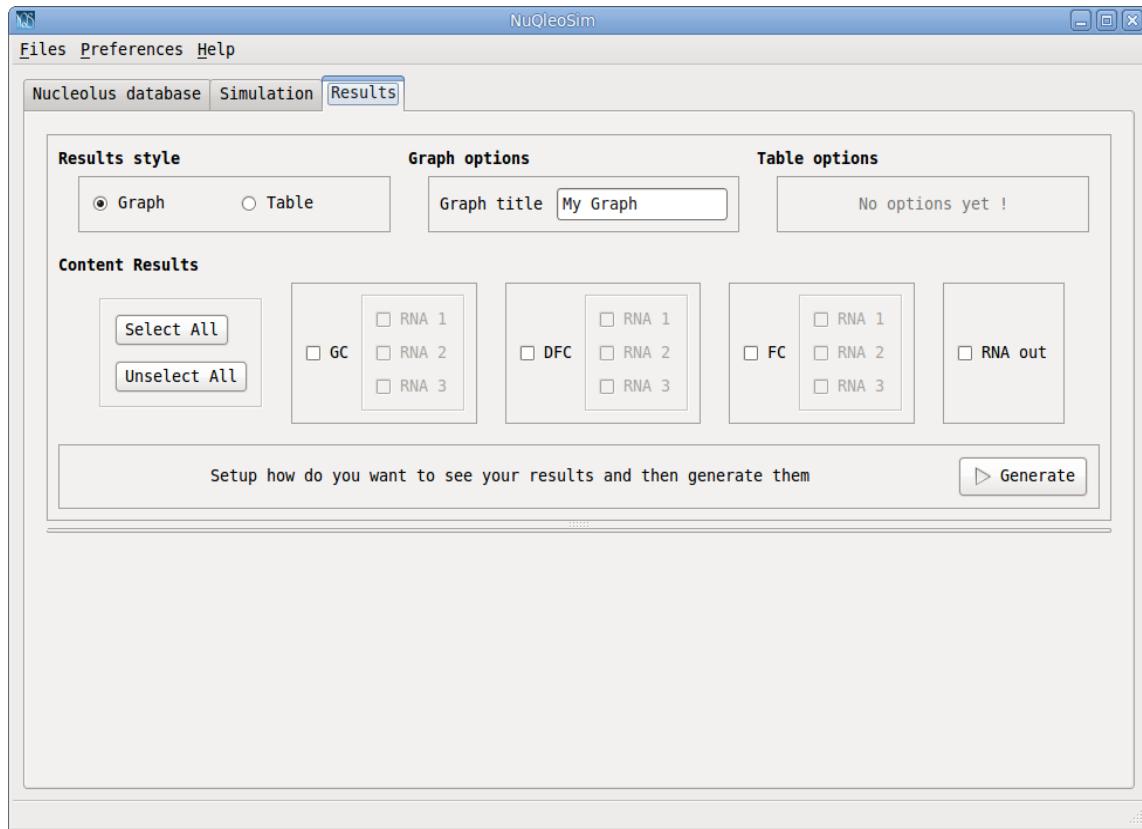


FIGURE 3.8 – *Onglet de paramétrage de la visualisation des résultats de la simulation.*

Lorsque la simulation est terminée le troisième et dernier onglet **résultat** est alors opérationnel. C'est-à-dire que l'utilisateur peut organiser les résultats issus de la simulation comme il le souhaite dans un ou divers graphes (Fig. 3.10) ou tableaux (Fig. 3.9). Afin de rendre l'interface la plus intuitive possible, la sélection des ARN à analyser se fait par compartiments. Chaque graphe ou tableau est alors généré dans un nouvel onglet, dans la partie basse de l'onglet, et peut être exporté aux formats PNG ou CSV par un simple clic sur le bouton approprié.

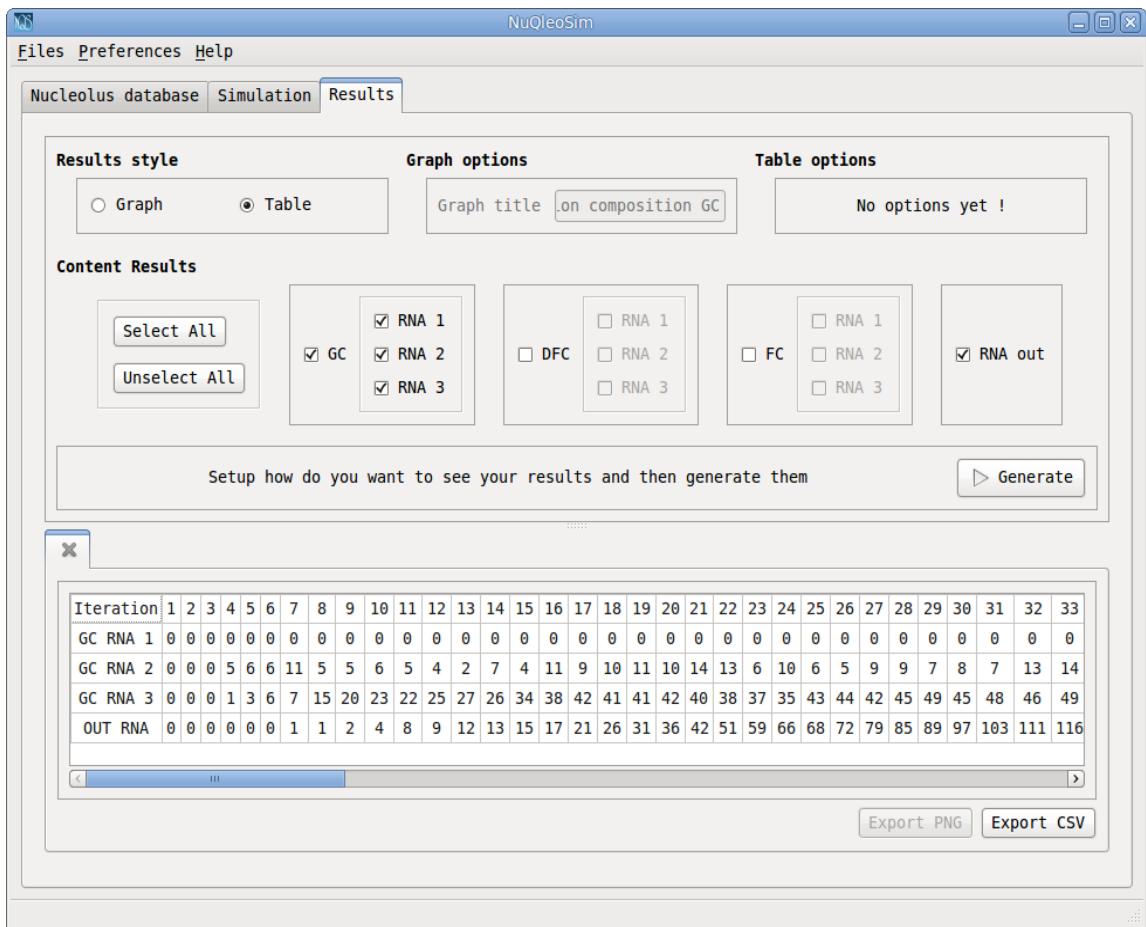


FIGURE 3.9 – Exemple de tableau de résultats.

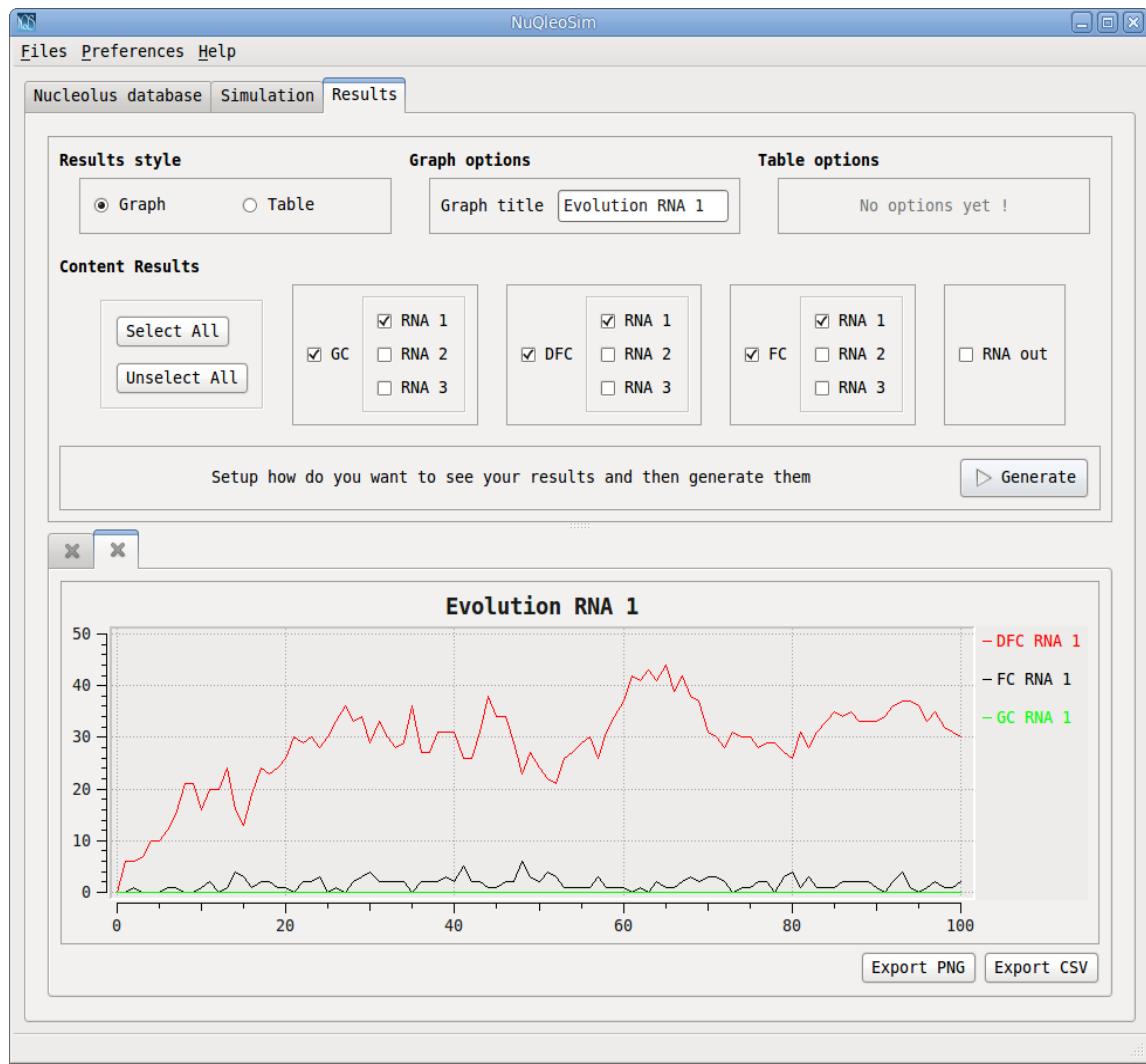


FIGURE 3.10 – Exemple de graphe de résultats.

4. Cas d'utilisation

Une fois **NuQleoSim** lancé l'utilisateur peut utiliser l'application de deux manières. Il peut soit consulter ou modifier la base de données moléculaire et expérimentale de **NuQleoSim**, soit paramétrier une simulation et la démarrer.

4.1 Communication avec la base de données

Lorsque l'utilisateur est sur l'onglet **Nucleolus Database**, il a accès à l'ensemble de la base de données de **NuQleoSim**. Sur la gauche de l'interface, se trouve une colonne répertoriant l'ensemble des molécules et expériences (respectivement dans deux listes différentes) de la base de données. En cliquant sur une molécule (respectivement une expérience), s'affichent sur la partie de gauche de l'interface, les informations la concernant (Fig. 3.3). Excepté le champ identifiant de la molécule (respectivement de l'expérience), tous les autres sont accessibles et modifiables à tout instant. En ce qui concerne l'ajout ou le retrait de molécules (respectivement expériences) de la base de données, il suffit de cliquer sur les boutons correspondants : en dessous de la liste en question, pour ajouter un élément et en bas à droite de l'interface, pour supprimer l'élément sélectionné à cet instant. Lors de l'ajout d'une molécule (respectivement une expérience) dans la base de données, l'identifiant dans cette base est attribué automatiquement, le reste des champs est vide et doit être renseigné par l'utilisateur. En particulier, il peut entrer l'identifiant de la molécule dans les bases de données proposées (PDB, UniProtKB et EMBL). Un simple clic sur le bouton à droite du champ rempli permet d'ouvrir un lien, dans le navigateur internet, vers la fiche de la molécule dans la base de données considérée. A la suppression d'une molécule (ou expérience), un message d'alerte apparaît pour prévenir l'utilisateur que toute suppression de la base de données est définitive.

4.2 Paramétrage et lancement d'une simulation

L'onglet **Simulation** permet à l'utilisateur de paramétrer une simulation sur le nucléole d'une cellule. Ce paramétrage se fait en deux temps. Dans un premier temps, il faut régler les paramètres portant sur l'environnement de la simulation, soit le nucléole. De gauche à droite dans la partie supérieure de l'interface, l'utilisateur peut préciser les trois dimensions de son nucléole, puis la durée d'une simulation (en nombre d'itérations ou de déplacements) et enfin le nombre de mesures qu'il veut faire au cours du temps de simulation. Dans la partie basse de **NuQleoSim** l'utilisateur peut sélectionner dans la liste des molécules de la base de données qui lui sont proposées, celles qu'il veut intégrer à sa simulation. Un simple clic sur la molécule dans la liste permet d'afficher les paramètres à configurer pour la simulation à gauche de la liste. Cependant, pour que ces modifications soient enregistrées et prises en compte dans la simulation la case en haut à gauche de ces paramètres doit être cochée. Une fois sélectionnée la molécule apparaît surlignée en rouge dans la liste. Une fois paramétrée à sa guise, l'utilisateur peut cliquer sur le bouton **Launch** pour démarrer la simulation.

A titre d'exemple, une simulation mettant en jeu deux protéines, dont les proportions sont précisées ci-après, va être présentée.

- INSULIN :
 - FC Concentration : 0.0 ; accessible
 - DFC Concentration : 0.0 ; accessible
 - GC Concentration : 2.0 ; accessible
 - Movement probability : 1
- :
 - FC Concentration : 0.0 ; non accessible
 - DFC Concentration : 0.0 ; non accessible
 - GC Concentration : 2.0 ; accessible
 - Movement probability : 1

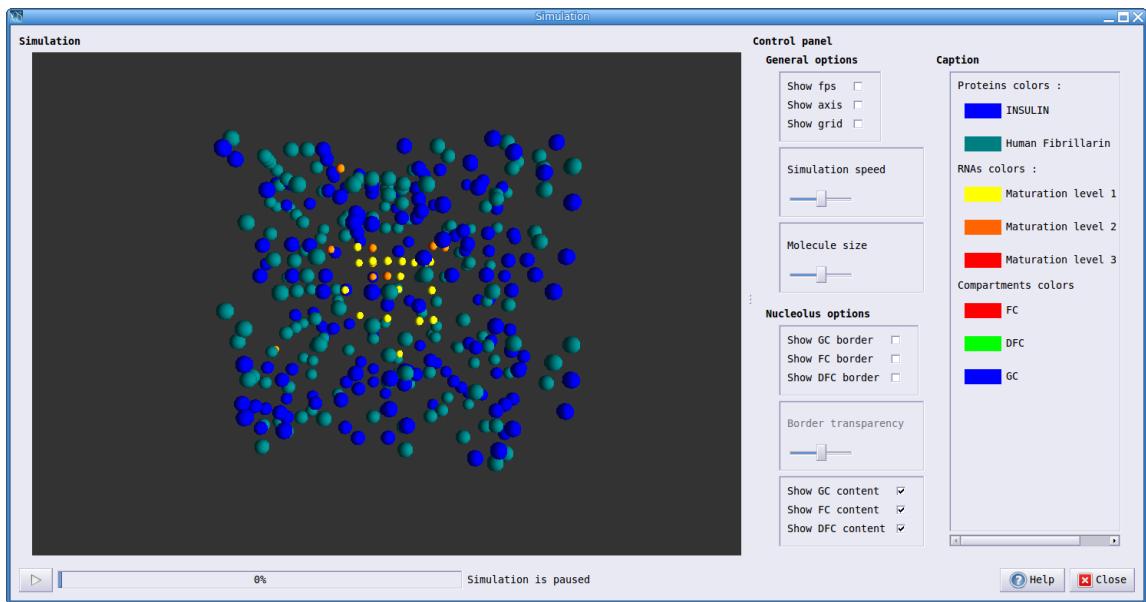


FIGURE 4.1 – Fenêtre de visualisation de la simulation à son commencement.

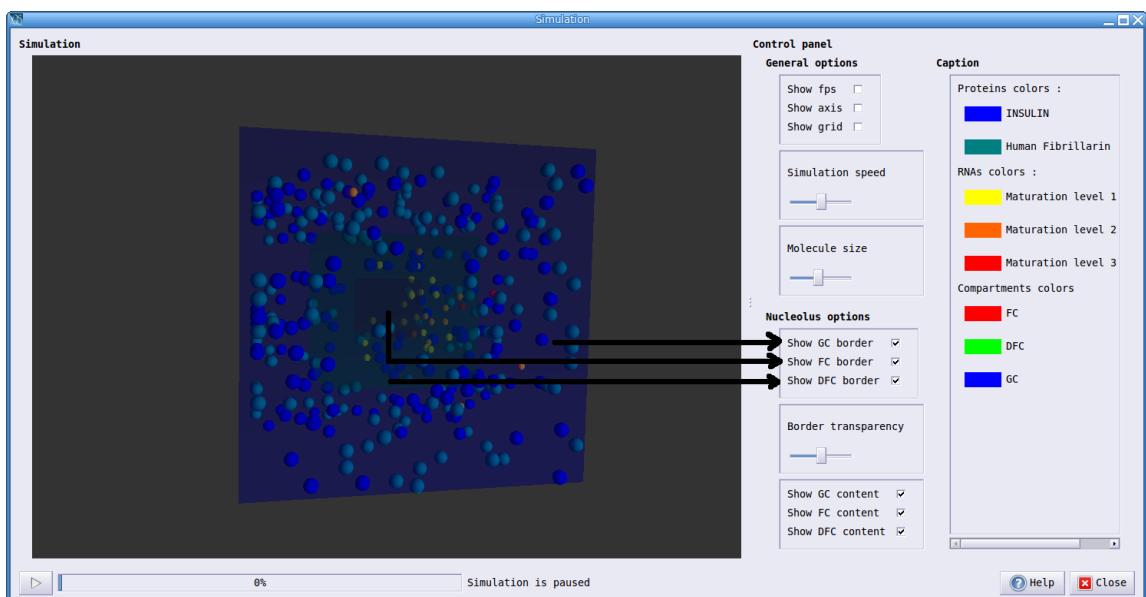


FIGURE 4.2 – Visualisation des composantes du nucléole

Au lancement de la simulation une nouvelle fenêtre (Fig. 4.1) s'ouvre dans laquelle se déroule la simulation dans un environnement tridimensionnel. La zone de vi-

sualisation proprement dite se trouve à gauche de la fenêtre qui s'est ouverte. À droite se trouve la légende des couleurs de la simulation ainsi que divers paramètres modulables par l'utilisateur.

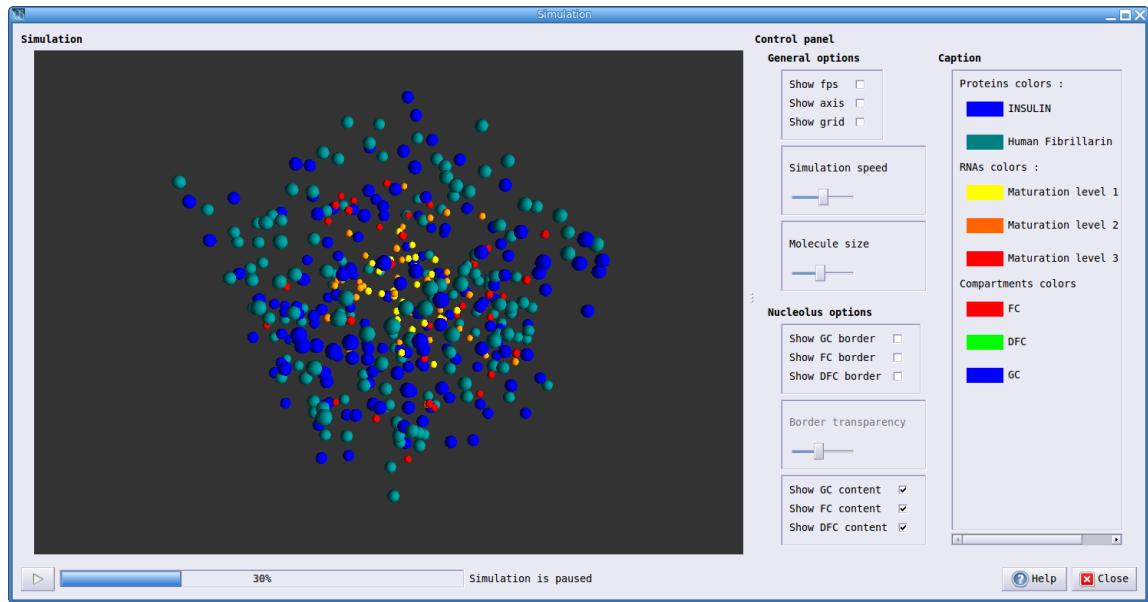


FIGURE 4.3 – Fenêtre de visualisation de la simulation terminée.

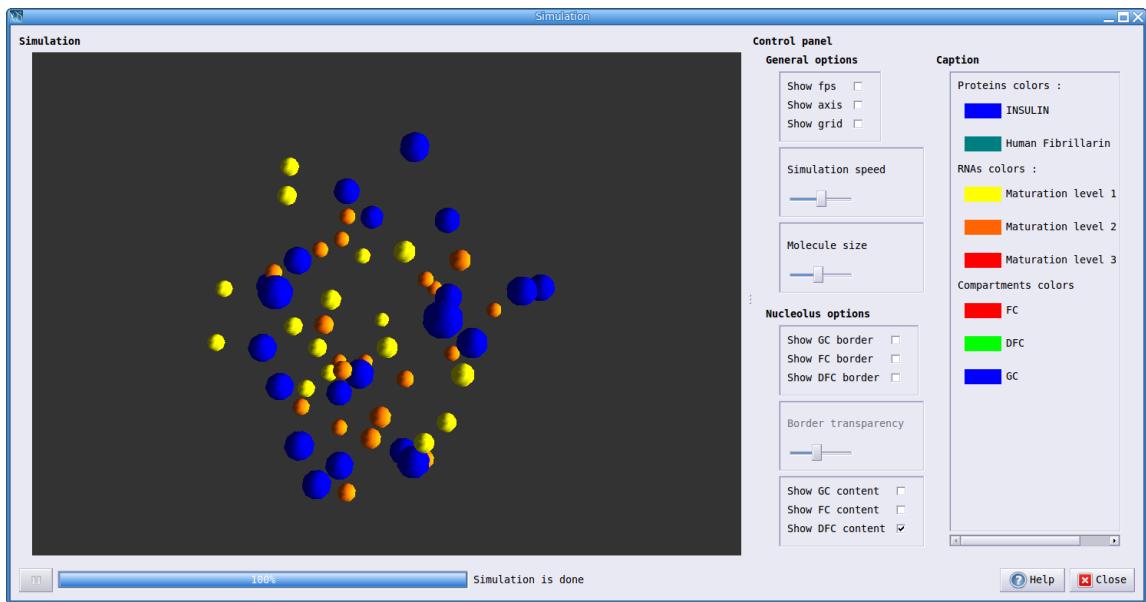


FIGURE 4.4 – *Zoom sur les molécules uniquement présentes dans le DFC une fois la simulation terminée.*

En premier lieu, l'utilisateur peut mettre la simulation en pause, grâce au bouton situé sous la simulation, pour pouvoir ajuster les divers paramètres de visualisation accessibles. En effet, la taille des molécules ainsi que la vitesse de la réaction peuvent être changés. De plus, il est possible de faire apparaître indépendamment en transparence les trois composantes du nucléole. Cette transparence est elle même ajustable par l'utilisateur. Il est également possible de ne visualiser que les molécules d'une ou plusieurs composantes ou de zoomer sur une zone d'intérêt (Fig. 4.4).

4.3 Présentation organisée des résultats

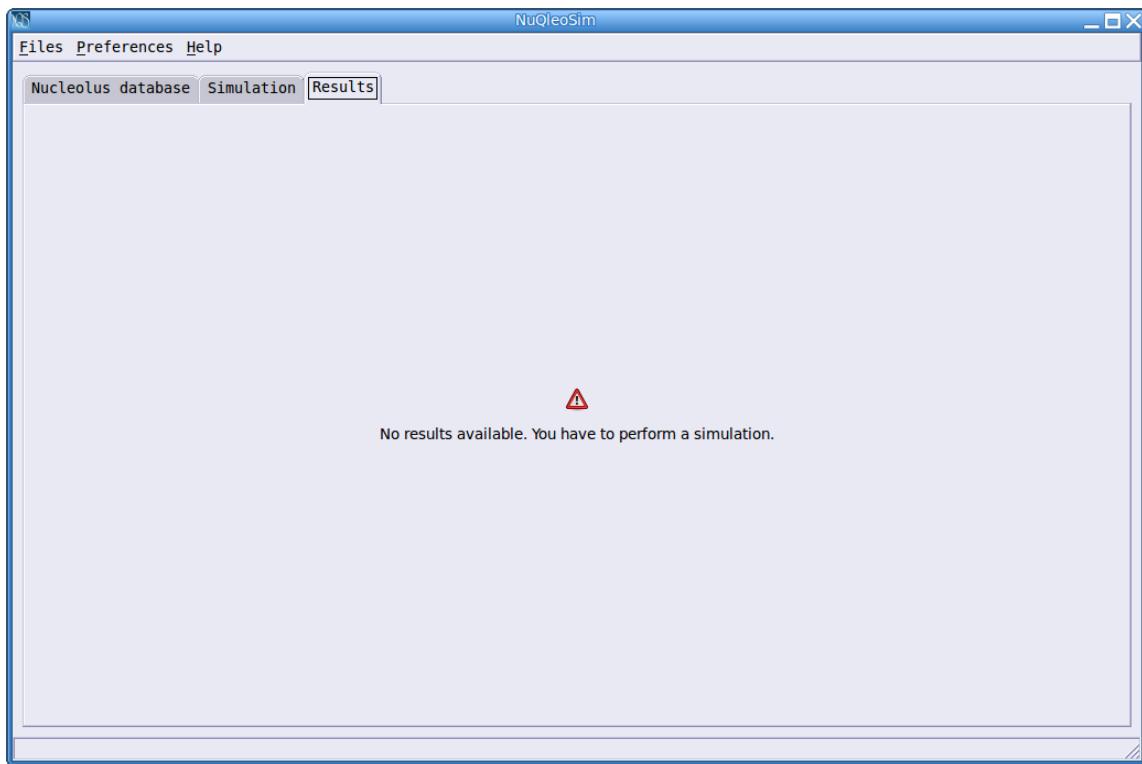


FIGURE 4.5 – L'onglet *Results* avant toute simulation.

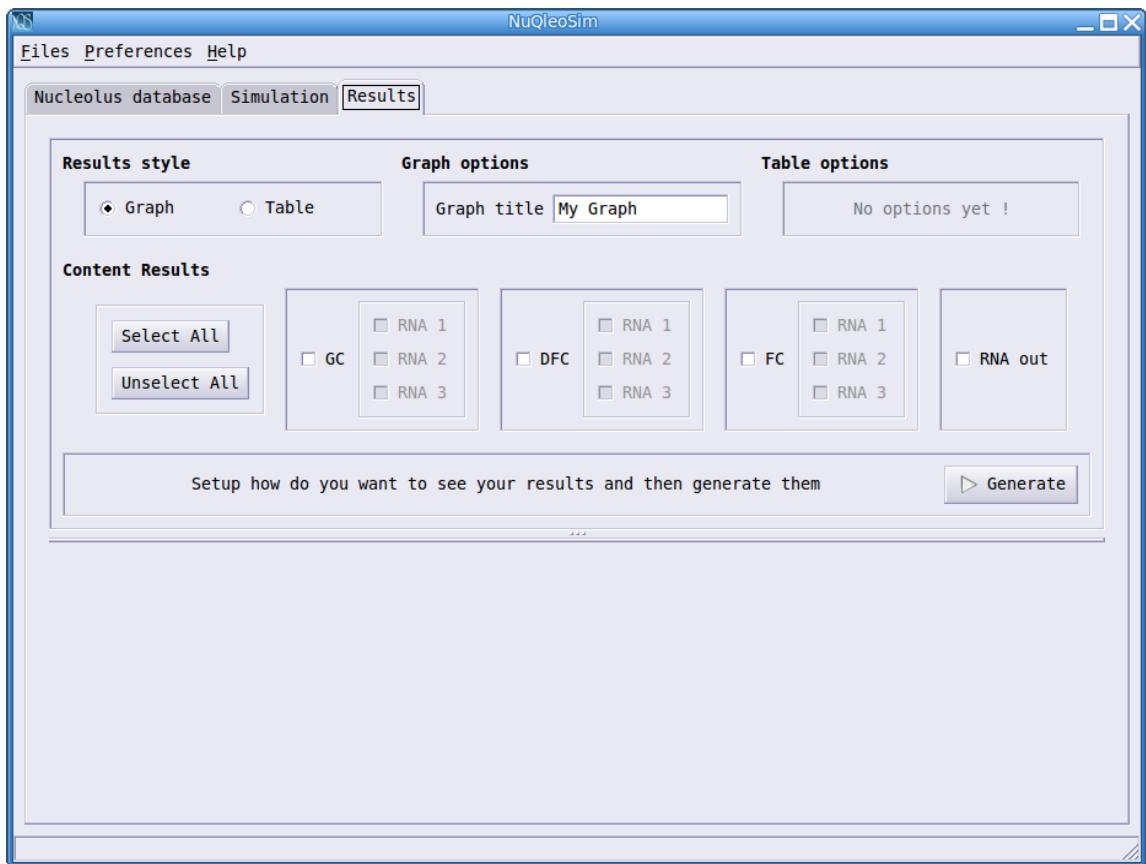


FIGURE 4.6 – L’onglet **Results** après une simulation : Paramétrage de l’analyse des résultats.

Tant qu’aucun résultat issu d’une simulation n’est disponible, l’onglet **Results** ne permet aucune action et affiche un message à l’utilisateur (Fig. 4.5). Lorsque la simulation est terminée (Fig. 4.3), le contenu de l’onglet **Results** est alors accessible (Fig. 4.6). L’utilisateur peut alors analyser les résultats de la simulation en les présentant sous forme de graphes ou de tableaux. Dans le cas d’un graphe, il peut le nommer puis sélectionner les molécules d’ARN qu’il veut analyser. Il en va de même pour le tableau. Pour générer le graphe (respectivement le tableau), l’utilisateur n’a qu’à appuyer sur le bouton **Generate** à droite. Dans la partie basse jusqu’alors inutilisée apparaît un onglet présentant le graphe généré (Fig. 4.7). Concernant la génération du tableau l’opération est la même (Fig. 4.8). L’utilisateur peut générer autant de graphe ou tableau qu’il le désire, ils s’inséreront dans de nouveaux onglets à côté du précédent (Fig. 4.8).

Pour terminer, si l'utilisateur veut conserver les résultats sous forme de graphes ou tableaux, ils peuvent être exportés au format CSV ou PNG (uniquement pour les graphes).

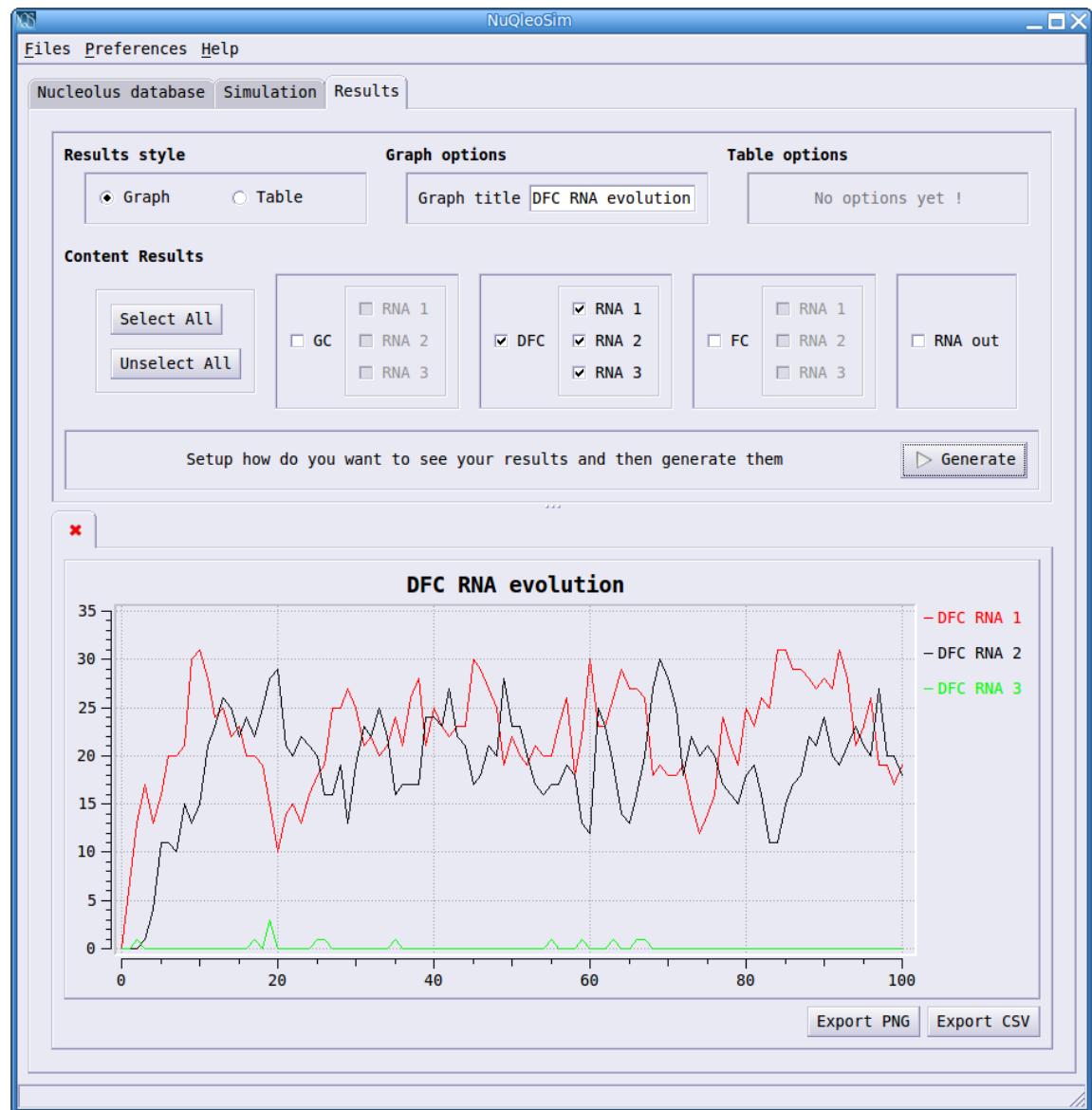


FIGURE 4.7 – Génération d'un graphe à partir des résultats de la simulation.

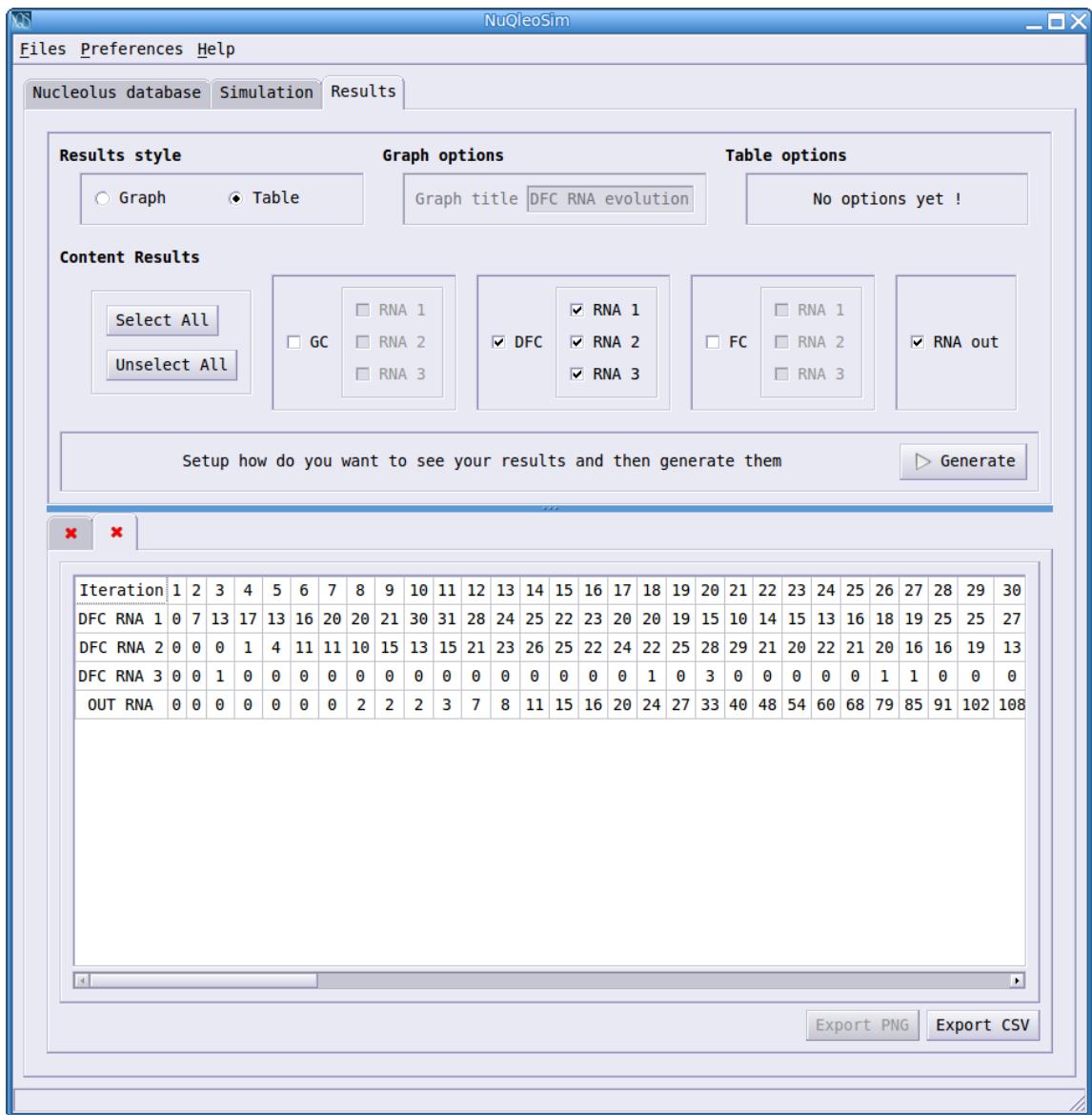


FIGURE 4.8 – Génération d'un tableau à partir des résultats de la simulation.

Conclusion

Le nucléole, domaine nucléaire dynamique, est le centre de synthèse des ribosomes. Son activité reflète l'équilibre entre le niveau de synthèse des ARN ribosomiaux (ARNr), l'efficacité de la maturation des ARNr et le transport des sous-unités ribosomiques vers le cytoplasme. Le nucléole est un domaine très dynamique dont l'apparence varie en fonction de son activité. En effet, quand la biogenèse des ribosomes est active, le nucléole est organisé en trois composantes majeures visibles en microscopie électronique : les centres fibrillaires (FC) sont des régions claires dont la taille varie de 0,1 à 1 μm ; ils sont entourés par le composant fibrillaire dense (DFC), dont le nom est directement lié à son aspect contrasté et à sa texture ; ces deux composants sont encastrés dans le composant granulaire (GC) constitué de granules de 15 à 20 nm. Le nucléole est un domaine nucléaire multifonctionnel qui joue un rôle important dans l'organisation nucléaire. Des mutations au niveau de gènes codant pour des protéines nucléolaires sont associées à des maladies humaines. C'est pourquoi il est actuellement envisagé de modéliser l'activité nucléolaire pour pouvoir comprendre ses mécanismes et apporter des pistes pour les études pathologiques.

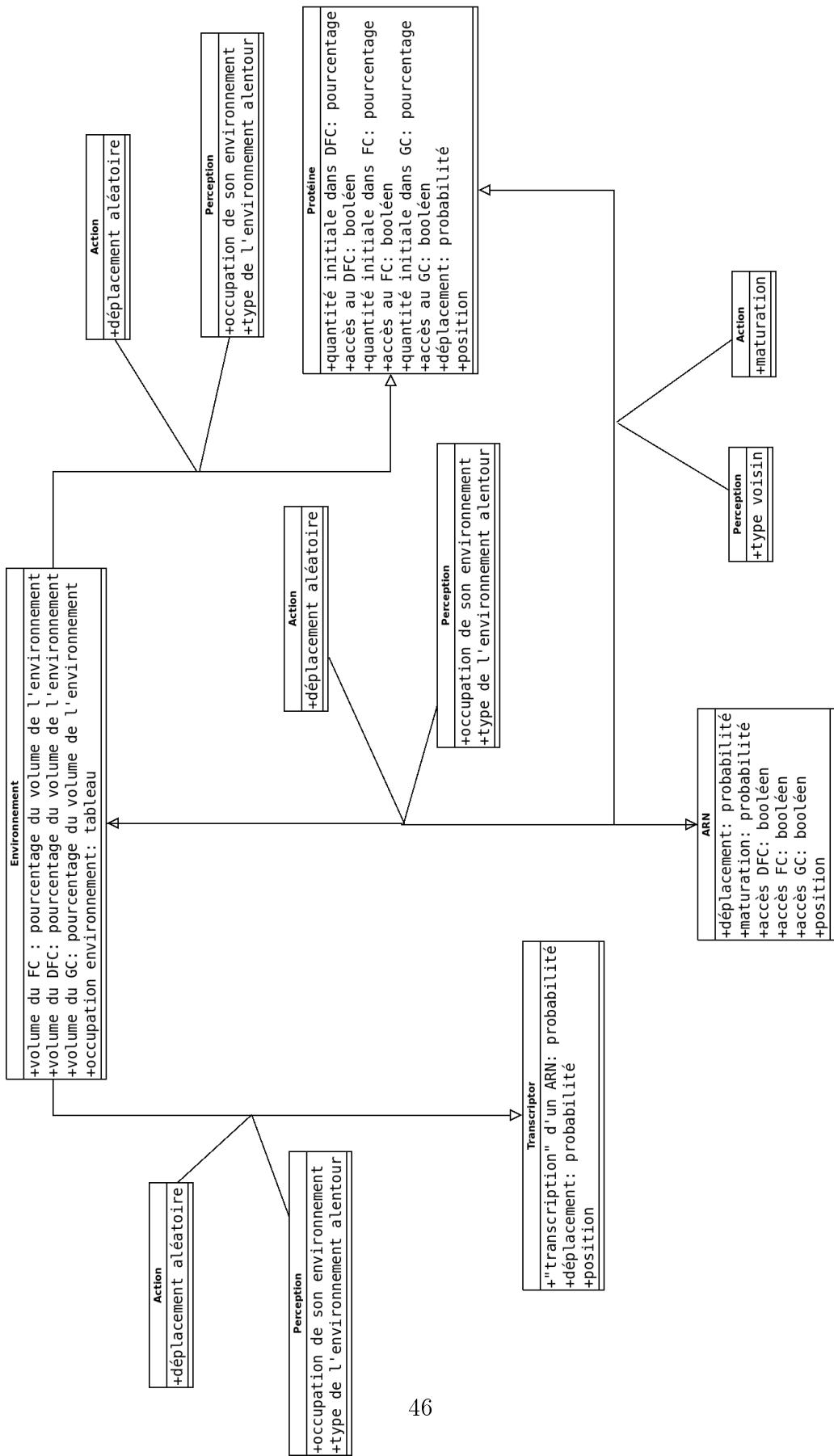
Le projet présenté permet de modéliser dans différentes conditions les diverses étapes de la biogenèse des ribosomes. **NuQleoSim** est une application comportant trois parties majeures. La première, la base de données, permet le recensement de molécules, protéines ou ARN. Ainsi l'utilisateur peut avoir accès aux différentes données générales sur la dite molécule mais aussi aux données qu'il aura lui-même consignées auparavant. Cette base de données peut donc être mise à jour par l'utilisateur dès que ce dernier le désire. De plus, il est possible d'accéder directement à l'ensemble des connaissances répertoriées sur la fiche de cette molécule dans une base de données internet via son identifiant dans la base de données en question. Dans la deuxième partie de l'application, l'utilisateur peut simuler les mouvements des molécules au sein d'un nucléole. Il peut pour cela définir le nucléole lui-même

ainsi que les molécules qui vont intervenir dans la simulation. Une fois la simulation terminée les résultats peuvent être affichées différemment sous forme de graphique ou de tableau grâce à la troisième et dernière partie.

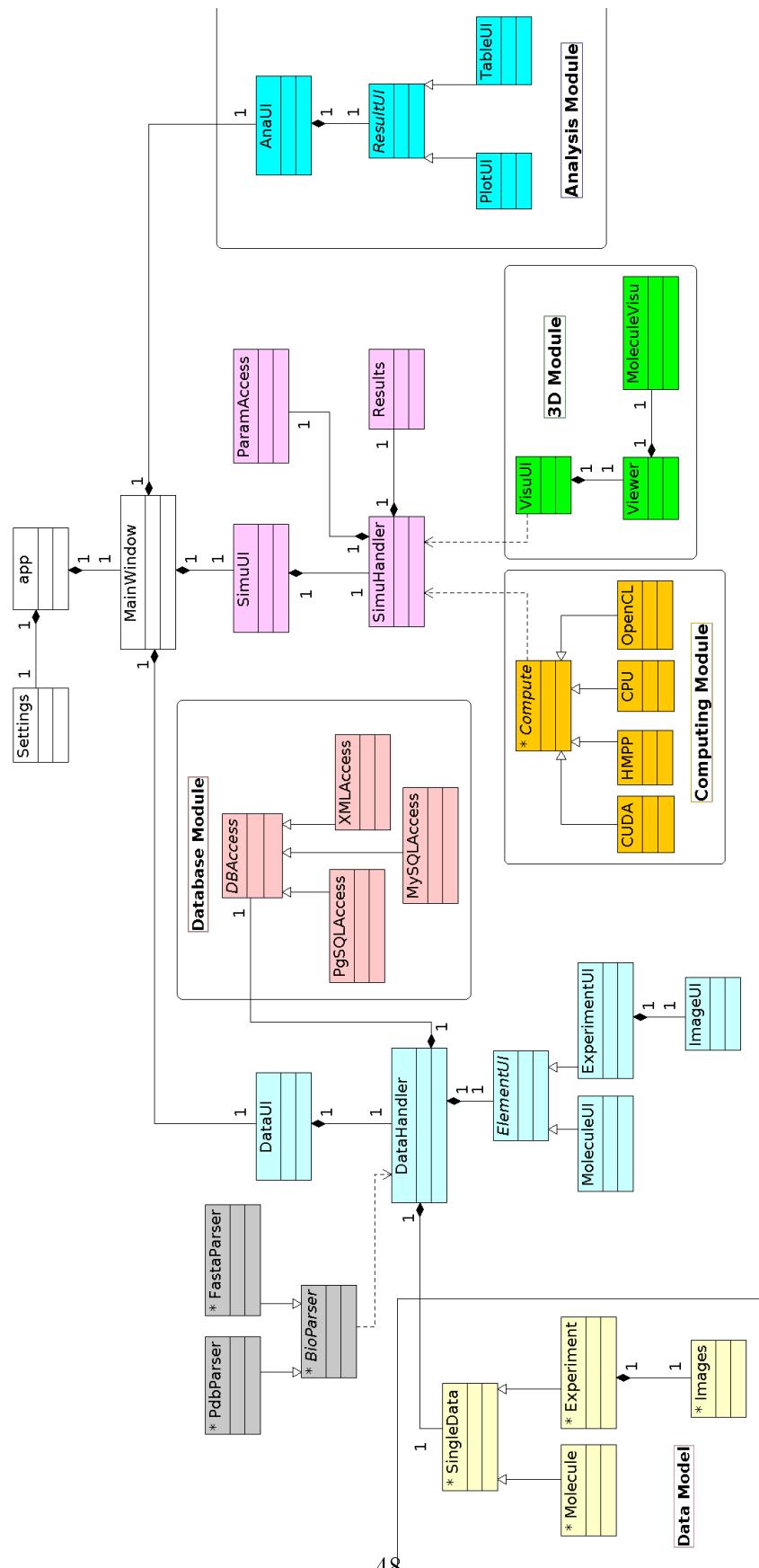
Néanmoins, de nouvelles améliorations peuvent être apportées au programme. En effet, la modularité du code permet d'ajouter aisément de nouveaux paramètres à la modélisation, mais aussi en parallèle dans l'interface, comme la spécification par l'utilisateur du rôle d'une molécule dans les étapes de maturations des ARNr. Il pourrait également être envisagé, à partir des résultats, de créer automatiquement une « fiche » de l'expérience simulée pour la conserver dans la base de données de **NuQleoSim**. Enfin, il pourrait être intéressant d'étendre le module de modélisation afin d'effectuer les calculs sur le GPU et ainsi gagner en performance.

Appendices

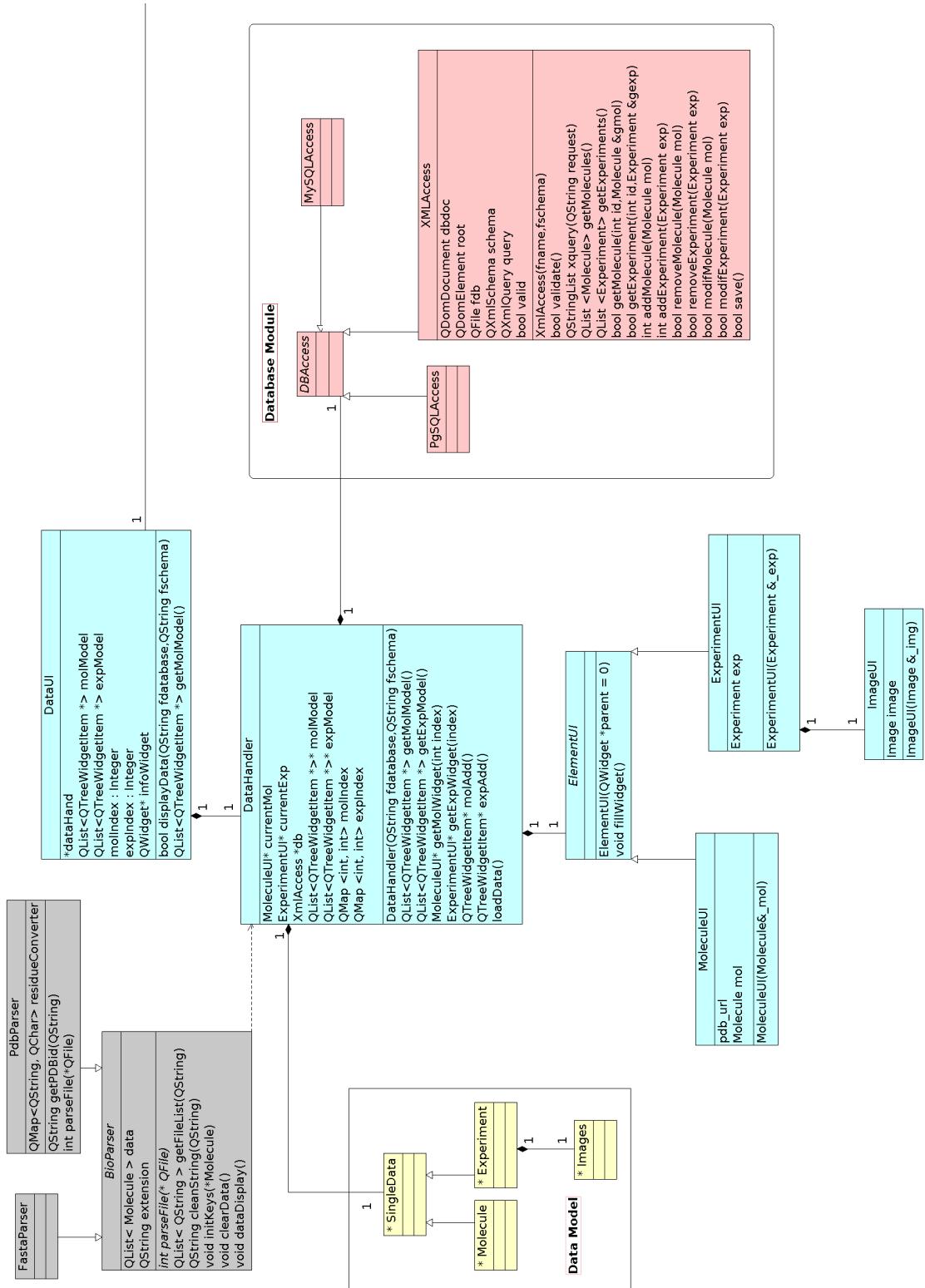
A. Diagramme du système multi agent



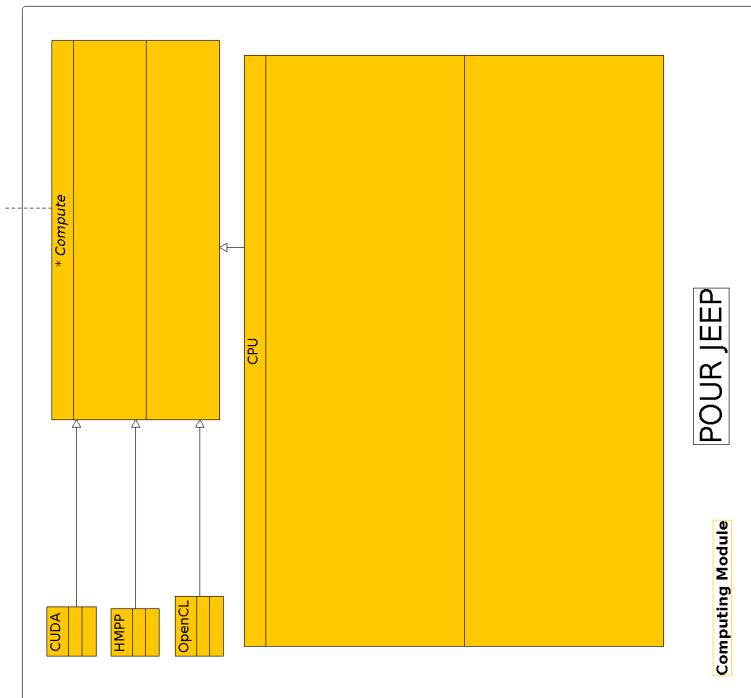
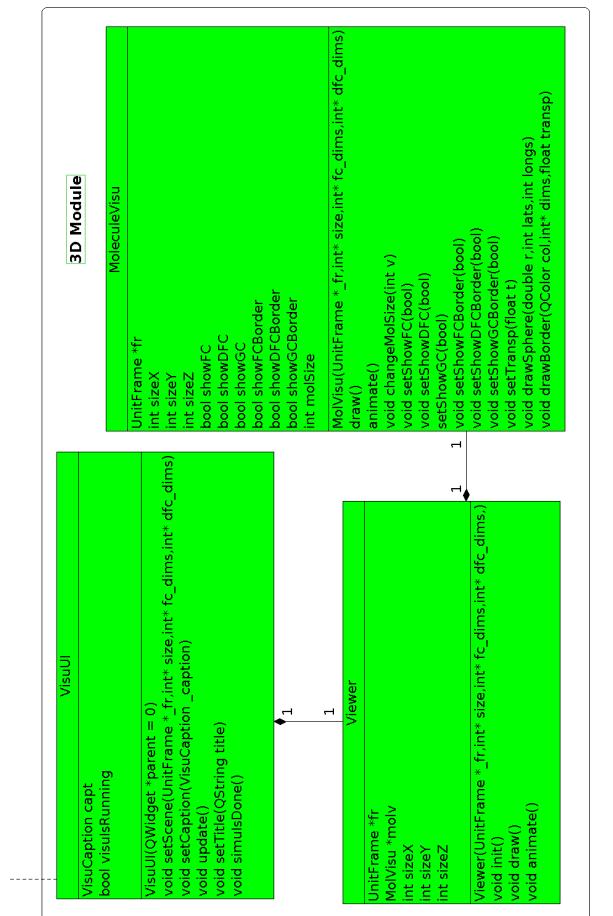
B. Vue générale du diagramme de classe

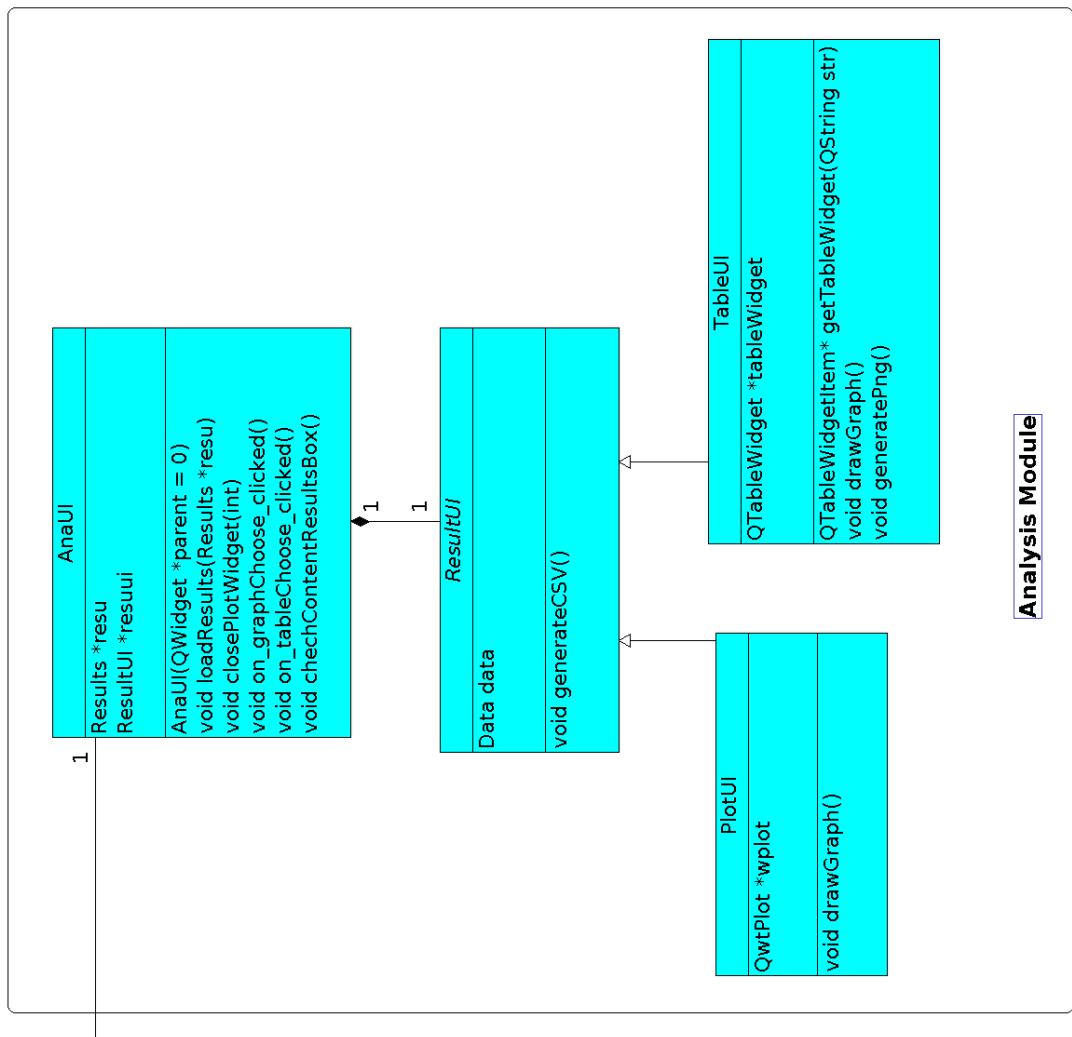


C. Diagramme de classe détaillé









Bibliographie

- [1] A.K.L. Leung, L. Trinkle-Mulcahy, Y.W. Lam, J.S. Andersen, M. Mann, and A.I. Lamond. Nopdb : nucleolar proteome database. *Nucleic Acids Research*, 34(suppl 1) :D218, 2006.
- [2] Ivan Raska, Peter J Shaw, and Dusan Cmarko. Structure and function of the nucleolus in the spotlight. *Curr Opin Cell Biol*, 18(3) :325–334, Jun 2006.
- [3] E. Sklar. Netlogo, a multi-agent simulation environment. *Artificial Life*, 13(3) :303–311, 2007.
- [4] J. Sprenger, J.L. Fink, S. Karunaratne, K. Hanson, N.A. Hamilton, and R.D. Teasdale. Locate : a mammalian protein subcellular localization database. *Nucleic acids research*, 2007.
- [5] Marc Thiry and Denis L J Lafontaine. Birth of a nucleolus : the evolution of nucleolar compartments. *Trends Cell Biol*, 15(4) :194–199, Apr 2005.