

Conception d'un Projet de recherche

FRECHE Arnaud, HÉRICÉ Charlotte, MOLA Saraï,
PAYSAN-LAFOSSE Typhaine, SANSEN Joris



Table des matières

1	Analyse	4
1.1	Contexte	4
1.2	État de l'existant	5
1.3	Analyse des besoins	8
2	Conception	10
2.1	Interface graphique	10
2.2	Création d'un nouveau réseau	10
2.3	Règles des gènes	11
3	Réalisation	12
3.1	Règles des gènes	12

Introduction

Chapitre 1

Analyse

1.1 Contexte

1.1.1 Sujet

Le métabolisme d'une cellule est un système complexe de transformations moléculaires et énergétiques qui se déroulent de manière ininterrompue dans la cellule et mettant en jeu un ensemble de réactions dites métaboliques. Ces réactions impliquent différents types de métabolites qui, suivant leurs positions dans la réaction, sont appelés substrats ou produits et catalysés par des enzymes.

La représentation visuelle des grandes fonctions métaboliques (glycolyse, photosynthèse ...) sous la forme de réseaux pourrait être un outil précieux pour faciliter l'avancement des travaux des chercheurs. Cette modélisation de réactions permet de réaliser des requêtes complexes comme, par exemple, le calcul (et la prédiction) de tous les métabolites pouvant être générés à partir d'un ensemble de composés sources.

Quelques logiciels disponibles internationalement permettent de travailler et d'automatiser l'étude de ces réseaux, tout en proposant des fonctionnalités telles que le calcul des modes élémentaires de flux ou la recherche de *minimal cut sets*¹. Cependant, ils peuvent être dépendants de logiciels non libres comme MATLAB ou alors ne pas avoir d'interface utilisateur conviviale, ce qui est le cas avec *regEfmtree* (logiciel sur lequel nous nous appuyerons pour ce projet).

1.1.2 Objectif

Dans le cadre de cette U.E., il nous est demandé de créer un programme (package et documentation à fournir à la fin du projet) nous permettant d'approfondir ou d'apprendre un langage de programmation, et de réaliser une analyse critique du travail effectué. L'objectif de ce projet sera donc de réaliser une interface graphique du programme *regEfmtree* qui ne peut être utilisé, à l'heure actuelle, que par des commandes à l'aide d'une console.

Deux possibilités s'offraient à nous pour réaliser cette interface : nous avons le choix entre les technologies Web et le langage Java. Nous avons choisi de nous appuyer sur les technologies Web. En effet, beaucoup de programmes utilisent une interface de type site Web leur permettant d'une part de créer une interface graphique conviviale, feuilles de style, tout en permettant une certaine modularité.

1. Un *minimal cut sets* [1] (MCS) est un ensemble minimal (irréductible) de réactions dans le réseau dont l'activation va certainement conduire à une défaillance de certaines fonctions du réseau.

1.2 État de l'existant

1.2.1 Efmtool

Efmtool [2] calcule les modes élémentaires de flux de réseaux métaboliques. Il est implémenté en Java et a été intégré à MATLAB.

Il a été développé par Marco Terzer. La version courante est la 4.7.1 (Décembre 2009).

1.2.2 RegEfmtool

RegEfmtool [3] est un outil informatique qui combine le calcul des modes élémentaires de flux et la régulation transcriptionnelle du réseau métabolique. Il a été développé, entre autres, par Christian Jungreuthmayer. Il a été créé afin d'accélérer le calcul de jeux complets de modes élémentaires de flux d'un réseau métabolique.

RegEfmtool est une extension d'Efmtool qui prend en compte la régulation transcriptionnelle des réseaux pour le calcul des modes élémentaires de flux.

La prise en compte de la régulation des gènes réduit de façon importante le nombre de solutions et permet d'éliminer constamment les modes qui ne peuvent exister biologiquement pendant et après le processus de calcul. Elle permet aussi de réduire considérablement le coût du calcul.

L'installation et l'utilisation de *regEfmtool* a été exclusivement testée sous Linux. Elle pourrait cependant fonctionner sous d'autres systèmes d'exploitation puisqu'il s'agit d'un programme Java. Il n'existe pas d'interface graphique de cette application, elle s'exécute donc en lignes de commandes via le terminal. La version courante de *RegEfmtool* est la 2.0 (Août 2012).

1.2.3 METATOOL

METATOOL [4] est un programme écrit en C développé de 1998 à 2000 par Thomas Pfeiffer (Berlin) en coopération avec Juan Carlos Nuno (Madrid), Stefan Schuster (Berlin) et Ferdinand Moldenhauer (Berlin).

Il sert à étudier la structure des réseaux métaboliques à partir d'équations de réactions stoechiométriques et permet notamment de calculer les modes élémentaires.

Les premières versions de METATOOL (jusqu'à la 4.9) ont été développées en C. Aujourd'hui, nous trouvons aussi une version de METATOOL en C++ mais cette version n'est pas au point. Dans la version actuelle (5.1) l'exécutable est désormais un module de MATLAB 7 et GNU 3.0 Octave, il se présente sous la forme d'un ensemble de fichiers scripts de MATLAB.

Les paramètres donnés en entrée pour le bon fonctionnement du logiciel METATOOL sont les suivants :

- La liste des réactions réversibles, ainsi que celle des réactions irréversibles, avec le nom des réactions,
- La liste des métabolites internes et externes impliqués dans les réactions,
- Les équations réactionnelles.

Le tout est rassemblé dans un fichier avec l'extension *.dat*

A la fin de son exécution, METATOOL a généré un fichier avec l'extension *.out* dans lequel se trouvent les résultats. Dans les versions de METATOOL écrites en C, le fichier de sortie contient l'ensemble des résultats sous forme de matrices, ainsi que des bilans qui permettent de décrire le réseau d'étude.

Les versions de METATOOL écrites en MATLAB produisent des résultats similaires en terme de calcul des matrices des modes élémentaires mais les résultats sont disposés différemment dans le fichier de sortie.

1.2.4 CellNetAnalyzer

CellNetAnalyzer [5] est un package de MATLAB (écrit en C) qui fournit un environnement compréhensible et convivial pour l'utilisateur et qui permet une analyse fonctionnelle et structurale de réseaux biochimiques. Il a été développé à l'institut Max Planck de Magdeburg par Steffen Klamt (depuis 2000) et Axel von Kamp (depuis 2007) notamment.

CellNetAnalyzer fournit une importante collection d'outils et d'algorithmes pour l'analyse structurale de réseaux.

C'est un programme gratuit pour une utilisation académique. Pour l'exécuter, il faut avoir installé MATLAB 7.0 ou une version ultérieure qui demande une licence. Il peut être utilisé sur Linux, Windows XP ou Mac.

Pour l'étude des modes élémentaires, CellNetAnalyzer fait appel à METATOOL via le logiciel MEX qui sert d'interface. MEX permet à MATLAB d'appeler tout logiciel C externe pour compléter les outils qu'il possède.

1.2.5 Yana

YANA [6] est un logiciel libre, écrit en JAVA, utilisant METATOOL pour l'étude des voies métaboliques. Il contient le logiciel METATOOL dans sa structure interne.

YANA [7] sert de façade et de sortie à METATOOL 6 tout en implémentant d'autres fonctions d'analyses du métabolisme (ex : quantification de l'activité enzymatique des réactions).

YANA s'occupe de l'entrée des données vers METATOOL puis il effectue une analyse syntaxique du fichier *.out* pour afficher les résultats sur une interface graphique et effectuer des analyses complémentaires sur ceux-ci.

1.2.6 Acom

ACoM [8] fonctionne à partir des fichiers de sortie de METATOOL. Il permet une classification automatique des modes élémentaires en fonction d'une taille minimale et d'un seuil de similarité. Il est surtout utilisé pour l'analyse des réseaux de grandes tailles où la manipulation des données à la main est laborieuse. ACoM est un programme C utilisable uniquement en mode console.

1.2.7 JACoMode

JACoMode est une interface Web permettant le lancement d'ACoM via le Web. En plus d'obtenir les résultats d'ACoM, JACoMode traite ces résultats pour obtenir d'autres résultats statistiques sur les modes élémentaires.

1.2.8 Langages

Notre projet nécessite l'utilisation d'une interface Web, dans ce cadre il existe :

- Mod Perl combiné avec Apache² et CGI³ mais la technologie utilisée est à l'heure actuelle dépassée
- Mod Python combiné avec Apache et CGI mais même remarque que précédemment
- CL-WHO avec Hunchentoot⁴ et ParenScript offre un bon environnement pour le développement Web

2. Apache HTTP Server

3. CGI : Common Gateway Interface.

4. Hunchentoot HTTP Server

- Java et ses applets⁵ apparaissent également comme un bon choix pour le développement Web
 - PHP⁶ couplé avec du JavaScript peut être un choix judicieux pour une application Web
- Parmi les différents choix précédemment cités, deux sortent du lot : Java et PHP avec leurs bibliothèques.

5. Applet Java : logiciel s'exécutant dans la fenêtre d'un navigateur Web (grâce à une machine virtuelle Java (JVM)), fournie aux utilisateurs sous la forme de bytecode Java.

6. PHP : Hypertext Preprocessor

1.3 Analyse des besoins

1.3.1 Besoins fonctionnels

Interface Homme Machine (IHM)

L'interface Web que nous allons réaliser devra permettre de charger la description d'un réseau pré-existant ou d'en créer un nouveau.

Elle donnera également le moyen à l'utilisateur de saisir les fonctions et options qui l'intéressent puis de lancer les calculs des modes élémentaires de flux du réseau d'intérêt.

Chargement des données

Une zone de chargement de fichiers à partir du disque dur de l'utilisateur sera présente sur l'interface.

Réglage des paramètres

Il sera possible pour un utilisateur confirmé ou habitué à l'interface d'avoir accès à un mode de réglage avancé des paramètres, s'il le désire. Ces derniers seront fixés à des valeurs par défaut pour les débutants.

Résultats

Enfin, la visualisation des résultats devra apparaître de façon claire et conviviale à l'utilisateur au travers de l'interface Web. De plus, les résultats pourront être exportés dans un fichier pour une analyse ultérieure.

Aide en ligne

Un message d'aide apparaîtra au survol du curseur de la souris sur la fonction ou l'option choisie. Ainsi l'utilisateur pourra avoir plus d'informations sur la commande concernée.

Lors de l'affichage de la page d'accueil, l'utilisateur aura le choix entre créer un réseau métabolique manuellement, ou bien le charger à partir de fichiers préexistant.

Création d'un nouveau réseau

Si l'utilisateur clique sur le bouton de création d'un nouveau réseau métabolique, une première page Web s'affichera. Il devra rentrer le nom des réactions ainsi que tous les métabolites et enzymes qui la composent. Il aura la possibilité de modifier une réaction (nom, enzymes, métabolites, coefficients stoechiométriques...). Dans le cas d'une validation, les données seront récupérées afin de générer automatiquement les fichiers nécessaires au fonctionnement du logiciel.

Chargement d'un réseau préexistant

Si l'utilisateur clique sur le bouton de chargement d'un réseau depuis la page d'accueil, il devra charger une série de fichiers (depuis le disque dur de son ordinateur) nécessaires au bon fonctionnement de *regEfmtool*.

Lancement du programme

Lorsque l'utilisateur aura créé son réseau manuellement ou l'aura chargé, il devra ensuite choisir les paramètres de calcul de *regEfmtree*. Nous avons fait le choix d'utiliser des cases à cocher en fonction de ce qu'il choisira. Pour les utilisateurs non expérimentés, les choix de base seront pré-sélectionnés. Il suffira ensuite de cliquer sur le bouton "Lancement" pour avoir les résultats générés par le logiciel.

Affichage des résultats

A revoir !!!

1.3.2 Besoins non fonctionnels

Portabilité

L'utilisation de *regEfmtree* s'appuie sur d'autres logiciels, nécessitant par exemple la version 1.7 de Java. De ce fait, ils devront être libre d'utilisation pour le secteur académique. L'interface devra être livrée avec tous les fichiers de configuration (pré-existants ou nouvellement créés) et indépendante du système d'exploitation.

Sécurité et robustesse

Il faudra gérer l'espace occupé par les fichiers chargés sur le serveur. Le site devra être stable et gérer au mieux les erreurs qui pourraient être générées lors du chargement des fichiers, de la modification des paramètres ou des calculs.

Documentation

L'écriture du code sera constituée de commentaires qui permettront la maintenance du code ainsi qu'une éventuelle amélioration de ce dernier par un tiers. L'interface Web créée, quand à elle, devra être fournie avec une documentation sur son installation, son utilisation, sa maintenance et une charte graphique déclarant les différents attributs du site (couleurs utilisées, police, logo, image,...).

Chapitre 2

Conception

2.1 Interface graphique

Comme nous l'avons précisé précédemment, notre interface graphique se présente sous la forme d'un site web. Lorsqu'on arrive sur le site, une page d'accueil s'affiche. Elle contient des informations relatives à l'utilisation du programme (à quoi il sert, que peut-on y faire ...). A partir de là, deux choix s'offrent à l'utilisateur :

- Soit il crée un nouveau réseau, en cliquant sur *Création* dans le menu situé en haut de page,
- Soit il charge un réseau qu'il a déjà créé.

Chaque page du site se présente de la même façon : un menu en haut de la page et le logo du site web dans un bandeau situé sur la gauche.

Le menu contient quatre onglets :

- *Accueil* qui permet de retourner sur la page d'accueil du site web
- *Création* qui permet de créer un nouveau réseau métabolique
- *Chargement* qui permet de charger un réseau pré-existant
- *Aide* qui permet de consulter une aide à l'utilisation du site.

On trouve aussi dans le menu, trois drapeaux qui permettent de changer la langue du site web. Les trois langues proposées sont : l'anglais, le français et l'allemand. Par défaut le site est en français.

2.2 Création d'un nouveau réseau

Si l'utilisateur choisi de créer un nouveau réseau, il arrive sur une page qui va lui permettre d'ajouter de nouvelles réactions à celui-ci.

Une première partie permet d'initialiser les fichiers dans le cas où l'utilisateur aurait déjà créé un réseau précédemment.

Une seconde partie permet de créer une nouvelle réaction. Celle-ci doit être de la forme *reaction* : $reag1 + reag2 \Rightarrow 2prod1 + 4prod2$.

L'utilisateur doit également préciser si la réaction est réversible ou non. Ensuite il clique sur le bouton *Ajouter*, la réaction est alors ajoutée au réseau et apparaît dans la zone de texte située en-dessous. Dans cette zone de texte, l'utilisateur a la possibilité de modifier les réactions déjà créées (mais ne peut pas en ajouter ou en supprimer car les règles de réversibilités ne seraient alors pas respectées pour l'écriture d'un fichier au format *.dat*). Il valide ensuite les modifica-

tions apportées en cliquant sur le bouton *Modifier*.

Enfin, l'utilisateur peut aussi générer un fichier au format *.dat* qui pourra être utilisé dans METATOOL par exemple.

Quand l'utilisateur a fini de créer toutes les réactions qui composent son réseau, il clique sur le bouton *Etape suivante* et arrive alors sur la page de création des règles des gènes.

2.3 Règles des gènes

Ces règles sont utilisées pour éliminer les modes élémentaires non possibles dans un réseau métabolique réel.

Une réaction R peut être 1-active ($R=1$), 0-active ($R=0$ ou encore full-active ($R=f$)).

La création des règles est régie selon certains principes :

- La réaction de sortie ne peut jamais être une réaction d'entrée.
On ne peut pas écrire : $R3 = (! (! 0R3) | 1R1)$
- Une réaction peut être utilisée plusieurs fois comme réaction d'entrée.
Ex : $R11 = ((! (! 0R3) | 1R5)) \& 1R3$
- le préfixe d'une réaction n'est valable que pour une opération.
Ainsi dans la règle $R11 = ((! (! 0R3) | 1R5)) \& 1R3$, la réaction R3 est 0-active dans l'opération OR et 1-active dans l'opération AND.
- Chaque réaction doit être entourée d'exactly une paire de parenthèses.
 $R3 = (! 0R1)$ est correcte, mais $R3 = ((! 0R1)$ et $R3 = ! 0R1$ sont incorrectes
- Il n'existe pas d'opération permettant de représenter la relation $R2 = R1$, elle sera représentée de la façon suivante : $R2 = (fR1 | fR1)$.

Tout d'abord, l'utilisateur choisit le nombre de réactions qui composent la règle qu'il veut créer. Une fois ce choix effectué, il clique sur le bouton *ok*. S'affichent alors de deux ou trois menus déroulant par réaction. Quand il y en a trois, le premier permet de choisir l'opérateur, le second la réaction et le dernier sa valeur (0, 1 ou f). Quand il y en a deux, ils correspondent à la réaction et à sa valeur. L'utilisateur doit entrer un nombre de réactions au moins égal à deux.

L'utilisateur ne peut sélectionner le nom d'une réaction que lorsqu'il a choisi la précédente. Ceci permet de ne proposer que les réactions non sélectionnées précédemment pour le choix de la réaction de sortie (principe utilisé dans la documentation de *regEfmtool*).

Enfin, quand l'utilisateur a sélectionné toutes les réactions ainsi que leur valeur, il clique sur le bouton *Ajouter*, la règle est alors écrite dans le fichier *generules.grfile* si l'utilisateur a bien rempli tous les champs. Celle-ci apparaît alors dans la zone de texte située en-dessous. Dans cette zone, l'utilisateur a la possibilité de modifier, ajouter ou supprimer une règle déjà écrite, mais également d'en écrire de nouvelles manuellement. lorsqu'il a fini ses modifications, il clique sur le bouton *Modifier*, le fichier *generules.grfile* est alors modifié en conséquence.

Une fois les règles entrées, l'utilisateur peut passer à l'étape suivante en cliquant sur le bouton correspondant. Il arrive alors sur la page de sélection des options nécessaires au lancement de la commande *regEfmtool*.

Chapitre 3

Réalisation

3.1 Règles des gènes

La page permettant la saisie des règles générales est obtenue à l'aide du fichier *generules.php*. L'affichage à l'ouverture de la page n'est composé que d'une zone de texte et d'un bouton *ok* qui est de type submit. Au clic, il fait appel à la fonction *add_reaction()* qui crée deux menus déroulant (pour la première et la dernière) ou trois menus déroulants (pour les autres) par réaction jusqu'à atteindre le nombre de réactions entrées par l'utilisateur. Elle vérifie également que le nombre de réactions entrées par l'utilisateur est au moins égal à 2, sinon elle affiche un message sous la forme d'une alert à l'utilisateur.

Les réactions sont récupérées à partir de *reactions.rfile*.

L'utilisateur peut sélectionner plusieurs fois la même réaction dans sa règle, sauf pour le choix de la dernière (ligne THEN). Cette particularité est gérée par la fonction *choice(form, val)*. Celle-ci permet de remplir les menus déroulants quand une réaction a été choisie et pour la dernière seules les réactions non sélectionnées précédemment apparaissent.

Lorsque l'utilisateur a choisi toutes ses réactions, leur opérateur et leur valeur, il clique sur le bouton *Ajouter*. Celui-ci fait appel à la fonction *validateForm()* qui vérifie que tous les champs ont bien été sélectionnés. Si la fonction retourne VRAI, il fait appel au fichier *createGrfile.php* qui écrit la règle dans le fichier *grfile.txt*.

Le fichier *createGrfile.php* écrit dans le fichier *grfile* selon certaines règles. En effet, l'écriture de la règle dépend des valeurs associées aux réactions et des opérateurs choisis.

Si la valeur de la réaction qui suit le "THEN" est 0, alors il y aura le symbole "!" au début de la règle (juste après le "="), sauf si la règle ne contient que deux réactions.

Si les autres réactions ont pour valeur :

- 0, alors on écrira (!0reac)
- 1, alors on écrira (!1reac)
- f, alors on écrira (!freac)

En revanche, si la valeur de la réaction après "THEN" est 1, les autres réactions s'écriront :

- 0reac si sa valeur est 0
- 1reac si sa valeur est 1
- freac si sa valeur est f

De plus, si l'opérateur "AND" est sélectionné il sera écrit sous la forme & dans le fichier, l'opérateur OR sera lui écrit |. Quand la règle est composée d'au moins trois réactions, des parenthèses sont ajoutées après chaque réaction sauf la première et la dernière. Des parenthèses entourent l'ensemble des réactions situées après le "=".

Pour mieux comprendre l'écriture du fichier *generules.grfile*, voici un exemple :
Choix de l'utilisateur sur la page web :
IF réaction : R1 valeur : 1
Opérateur : AND réaction : R2 valeur : 0
Opérateur : OR réaction : R3 valeur : 0
THEN réaction : R4 valeur : 0

Règle écrite dans le fichier :
R4 = (!((1R1 & (!0R2)) | (!0R3))

Bibliographie

- [1] <http://bioinformatics.oxfordjournals.org/content/20/2/226>.
- [2] <http://www.csb.ethz.ch/tools/efmtool>.
- [3] Christian Jungreuthmayer, David E. Ruckerbauer, and Jürgen Zanghellini. Utilizing gene regulatory information to speed up the calculation of elementary flux modes. 2012.
- [4] Axel von Kamp and Stefan Schuster. Metatool 5.0 : fast and flexible elementary modes analysis. *Bioinformatics*, August 2006.
- [5] <http://www.mpi-magdeburg.mpg.de/projects/cna/cna.html>.
- [6] Roland Schwarz, Patrick Musch, Axel von Kamp, Bernd Engels, Heiner Schirmer, Stefan Schuster, and Thomas Dandekar. Yana – a software tool for analyzing flux modes, gene-expression and enzyme activities. *BMC Bioinformatics*, June 2006.
- [7] <http://yana.bioapps.biozentrum.uni-wuerzburg.de/>.
- [8] S. Pérès, F. Vallée, M. Beurton-Aimar, and J.P. Mazat. Acom : A classification method for elementaryfluxmodes based on motif finding. *Biosystems*, March 2011.