



UNIVERSITÉ DE BORDEAUX

COMMUNICATION ET CONCEPTION D'UN PROJET DE
RECHERCHE ET/OU DÉVELOPPEMENT

Cahier des charges

Arnaud FRÈCHE
Charlotte HÉRICÉ
Sarai MOLA
Typhaine PAYSAN-LAFOSSE
Joris SANSEN

Mme. Marie
BEURTON-AIMAR

Table des matières

Introduction	2
1 Contexte	3
1.1 Sujet	3
1.2 Objectif	3
2 État de l'existant	4
2.1 Efmtree	4
2.2 <i>RegEfmtree</i>	4
2.3 METATOOL	4
2.4 CellNetAnalyzer	5
2.5 Yana	5
2.6 Acom	5
2.7 JACoMode	6
2.8 Langages	6
3 Besoins fonctionnels et non fonctionnels	7
3.1 Besoins fonctionnels	7
3.1.1 Interface Homme Machine (IHM)	7
3.1.2 Chargement des données	7
3.1.3 Réglage des paramètres	7
3.1.4 Résultats	7
3.1.5 Aide en ligne	7
3.1.6 Maquette de l'interface	8
3.2 Besoins non fonctionnels	12
3.2.1 Portabilité	12
3.2.2 Sécurité et robustesse	12
3.2.3 Temps de calcul	12
3.2.4 Documentations	13
4 Choix et justifications	14
4.1 Langages	14
4.2 Accessibilité	14
Bibliographie	14

Introduction

Le métabolisme correspond à l'ensemble des processus complexes et incessants de transformation de matière et d'énergie par la cellule ou l'organisme, au cours des phénomènes d'édification et de dégradation organiques (anabolisme et catabolisme). Ces molécules, appelées aussi métabolites, et les réactions dans lesquelles ils interviennent forment des réseaux métaboliques.

Analyser les réseaux métaboliques peut parfois s'avérer complexe étant donné l'importance de la taille de certains. L'outil bioinformatique devient donc vite indispensable dans le traitement de telles données.

Le but de notre projet sera donc de mettre en place une interface graphique pour le logiciel *regEfmtree* [1], qui est un outil de calcul des modes élémentaires de flux. Un mode élémentaire est un chemin métabolique au sein d'un réseau, c'est à dire un jeu de réactions uniques pour lequel, à l'état stationnaire, le flux global à travers le processus est nul.

La création de cette interface aura pour but de rendre l'utilisation de *regEfmtree* plus conviviale et de faire en sorte que les résultats générés soient visuellement interprétables.

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 *regEfmtool* (logiciel sur lequel nous nous appuyerons pour ce projet).

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 *regEfmtool* 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* [2] (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.

État de l'existant

2.1 Efmtool

Efmtool [3] 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).

2.2 *RegEfmtool*

RegEfmtool [4] 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).

2.3 METATOOL

METATOOL [5] 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.

2.4 CellNetAnalyzer

CellNetAnalyzer [6] 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 structurelle 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 structurelle 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.

2.5 Yana

YANA [7] 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 [8] 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.

2.6 Acom

ACoM [9] 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.

2.7 JACoMode

JACoMode est une interface Web permettant 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.

2.8 Langages

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

- Mod Perl combiné avec Apache¹ et CGI² 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 Hunchentoot³ et ParenScript offre un bon environnement pour le développement Web
- Java et ses applets⁴ apparaissent également comme un bon choix pour le développement Web
- PHP⁵ 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.

1. Apache HTTP Server

2. CGI : Common Gateway Interface.

3. Hunchentoot HTTP Server

4. 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.

5. PHP : Hypertext Preprocessor

Besoins fonctionnels et non fonctionnels

3.1 Besoins fonctionnels

3.1.1 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, mais également de modifier les descriptions de ce dernier.

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. De plus, un script récurrent de commandes pourra être enregistré et chargé par la suite.

3.1.2 Chargement des données

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

Selon le type du fichier donné en paramètre d'entrée, une liste de choix possibles pourra être proposée. Ainsi les résultats d'expériences similaires (suppression, modification des réactifs, produits ou enzymes du réseau) devront pouvoir être comparés avec un affichage en vis-à-vis, ce qui implique une sauvegarde temporaire des résultats sur le serveur.

3.1.3 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.

3.1.4 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. Par ailleurs, nous allons essayer de mettre en place un dispositif d'annotations des fichiers.

3.1.5 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.

3.1.6 Maquette de l'interface

Nous avons réalisé une série de maquettes pour avoir une idée du genre d'interface Web que nous voudrions. Elles ont été réalisées avec le logiciel Pencil [10] et permettent de mieux illustrer le rendu final de notre projet.

Page d'accueil

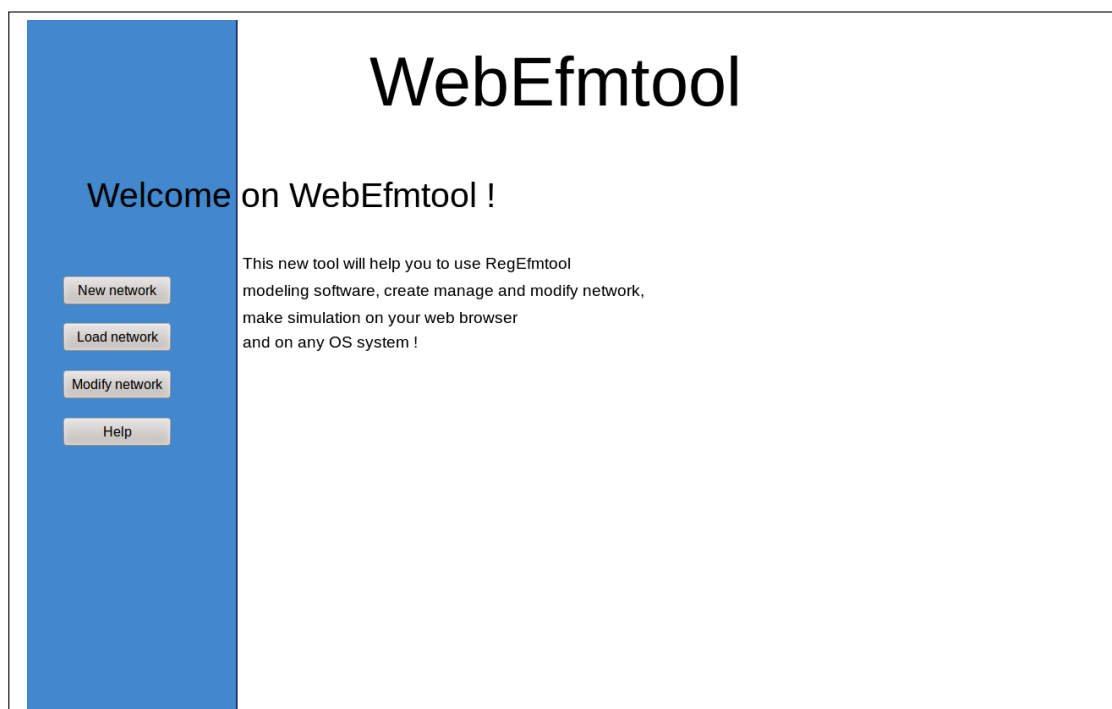


FIGURE 3.1 – Page d'accueil

Le logiciel *regEfmtool* étant en anglais, nous avons choisi d'utiliser cette même langue pour notre interface. De plus, cela permettra son utilisation par un plus grand nombre d'utilisateurs.

Lors de l'affichage de la page d'accueil (Figure 3.1), 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 (Figure 3.2), une première page Web s'affichera. Il devra rentrer les noms de tous les métabolites et enzymes participant aux réactions, ainsi que celui du réseau. Il aura la possibilité de valider ou supprimer chaque métabolite et enzyme. Dans le cas d'une validation, les données seront récupérées afin de générer automatiquement les fichiers nécessaires au fonctionnement du logiciel.

WebEfmtool

Create network

network name :

list of metabolites :

metabolite 1	<input type="button" value="ok"/>	<input type="button" value="X"/>
metabolite 2	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>

list of enzymes :

enzyme 1	<input type="button" value="ok"/>	<input type="button" value="X"/>
enzyme 2	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>

list of co-factors :

co-factor 1	<input type="button" value="ok"/>	<input type="button" value="X"/>
co-factor 2	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>
<input type="text"/>	<input type="button" value="ok"/>	<input type="button" value="X"/>

FIGURE 3.2 – Entrée des composants du réseau

WebEfmtool

Create Reactions

reaction name :

reagents of the reaction	stoichiometry
<input type="text" value="reagent"/>	<input type="text" value="1.25"/>
<input type="text" value="reagent"/>	<input type="text" value="1.25"/>
<input type="text" value="reagent"/>	<input type="text" value="1.25"/>
<input type="button" value="add more"/>	

enzymes
<input type="text" value="enzyme"/>
<input type="button" value="add more"/>
Co-factors
<input type="text" value="co-factor 1"/>
<input type="text" value="1.25"/>
<input type="button" value="add more"/>

products of the reaction	stoichiometry
<input type="text" value="product 1"/>	<input type="text" value="1.25"/>
<input type="text" value="product 2"/>	<input type="text" value="1.25"/>
<input type="text" value="product 3"/>	<input type="text" value="1.25"/>
<input type="button" value="add more"/>	

reversibility : ☐ yes ☐ no

FIGURE 3.3 – Création des réactions

Une fois les noms des composants du réseau métabolique entrés, l'utilisateur va devoir créer les réactions une à une (Figure 3.3). Tout d'abord, il devra s'occuper des réactifs de la réaction : un menu déroulant lui donnera le choix entre tous les métabolites qu'il a saisi précédemment et il n'aura plus qu'à entrer le coefficient stœchiométrique associé. Il en sera de même pour les enzymes (sans le coefficient de stœchiométrie dans ce cas), les produits et les co-facteurs. Pour ce qui est de la réversibilité des réactions, il suffira de cocher la bonne case. L'utilisateur pourra ensuite ajouter une réaction ou valider son réseau.

Chargement d'un réseau préexistant

The screenshot shows a web application titled 'WebEfmtool' with a sidebar on the left containing buttons for 'Create network', 'Load network', 'Modify network', and 'Help'. The main area is titled 'Load a network' and contains six rows of file upload fields, each with a text input and a 'browse' button. The fields are labeled: 'metabolites file', 'reactions file', 'reversibility file', 'stoichiometry file', 'reactions reversibility file', and 'genome file'. At the bottom of the main area are two large buttons: 'return' and 'validate'.

FIGURE 3.4 – Chargement du réseau

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*. Les fichiers doivent être dans le format adéquat et un message d'erreur apparaîtra si ce n'est pas le cas. Au cas où, une fonction d'aide sera disponible pour avoir un modèle de fichiers à charger.

Lancement du programme

Lorsque l'utilisateur aura créé son réseau manuellement ou l'aura chargé, il devra ensuite choisir les paramètres de calcul de *regEfmtool*. 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 "Run" pour avoir les résultats générés par le logiciel. Nous n'avons pas représenté toutes les options du lancement de *regEfmtool* sur cette maquette, pour des raisons d'affichage.

WebEfmtool

Create network

Load network

Modify network

Help

Run WebEfmtool

Simulation options

Results display

☐ on WebEfmtool
☐ save in a text file

Kind of stoichiometric file :

☐ flux-analyser-dir
☐ flux-analyser-files
☐ reaction-list
☐ excel
☐ stoichiometry
☐ sbml

Output options :

☐ null
☐ count
☐ text-boolean
☐ text-doubles
☐ binary-boolean
☐ binary-doubles
☐ matlab

Results view options :

☒ metabolites
☒ reactions
☒ unique flux
☒ iterations
☒ stoich matrix
☒ kernel matrix

Run

FIGURE 3.5 – Lancement de RegEfmtool

Affichage des résultats

Nous proposons dans la Figure 3.6 un exemple de ce que verra l'utilisateur après avoir lancé le logiciel. Il y aura une série de "boîtes" dans lesquelles seront affichés les différents éléments générés par *regEfmtool*. Il sera également possible de lancer une comparaison des résultats entre deux réseaux en appuyant sur le bouton "Compare".

WebEfmtool

Create network

Load network

Modify network

Help

Run WebEfmtool

Results

Network metabolites :

M_13dpg_c, M_2pg_c, M_3pg_c, M_6pgc_c, M_6pgl_c, M_ac_c, M_ac_b, M_acald_c, M_accoa_c, M_acon_C_c, M_actp_c, M_adp_c, M_akg_c, M_amp_c, M_ara_L_c, M_ara_L_b, M_atp_c, M_atpmain_b, M_cit_c, M_co2_c, M_co2_b, M_coa_c, M_d6pg_c, M_dhap_c, M_e4p_c, M_eto_c, M_eto_b, M_f6p_c, M_fdp_c, M_for_b, M_for_c, M_fru_b, M_fru_c, M_fum_b, M_fum_c, M_g3p_c, M_g6p_c, M_gal_c, M_gal_b, M_glc_D_b, M_glc_c, M_h2_b, M_h2_c, M_hic_c, M_iac_D_c, M_iac_D_b, M_m6p_c, M_mal_L_c, M_mal_L_b, M_man_b, M_nad_c, M_nadh_c, M_nadp_c, M_nadph_c, M_nh4_c, M_nh4_b, M_o2_b, M_o2_c, M_oaa_c, M_peg_c, M_pyr_c, M_pyr_b, M_g8_c, M_g8h2_c, M_f5p_c, M_ru5p_D_c, M_s7p_c, M_succ_c, M_succ_b, M_succoa_c, M_xu5p_D_c, M_xylo_c, M_xylo_b, M_xyly_c,

Network reactions :

R_PG1: M_g6p_c <=> M_f6p_c
R_PFK: M_atp_c + M_f6p_c -> M_adp_c + M_fdp_c
R_FBP: M_fdp_c -> M_f6p_c
R_FBA: M_fdp_c <=> M_dhap_c + M_g3p_c
R_TPI: M_dhap_c <=> M_g3p_c
R_GAPD: M_g3p_c + M_nad_c <=> M_13dpg_c + M_nadh_c
R_PGK: M_3pg_c + M_atp_c <=> M_13dpg_c + M_adp_c
R_PGM: M_2pg_c <=> M_3pg_c
R_ENO: M_2pg_c <=> M_peg_c
R_PVK: M_adp_c + M_peg_c -> M_atp_c + M_pyr_c
R_PPS: M_atp_c + M_pyr_c -> M_amp_c + M_peg_c

Compare

FIGURE 3.6 – Exemple d'affichage des résultats pour *E. Coli*

Comparaison des résultats

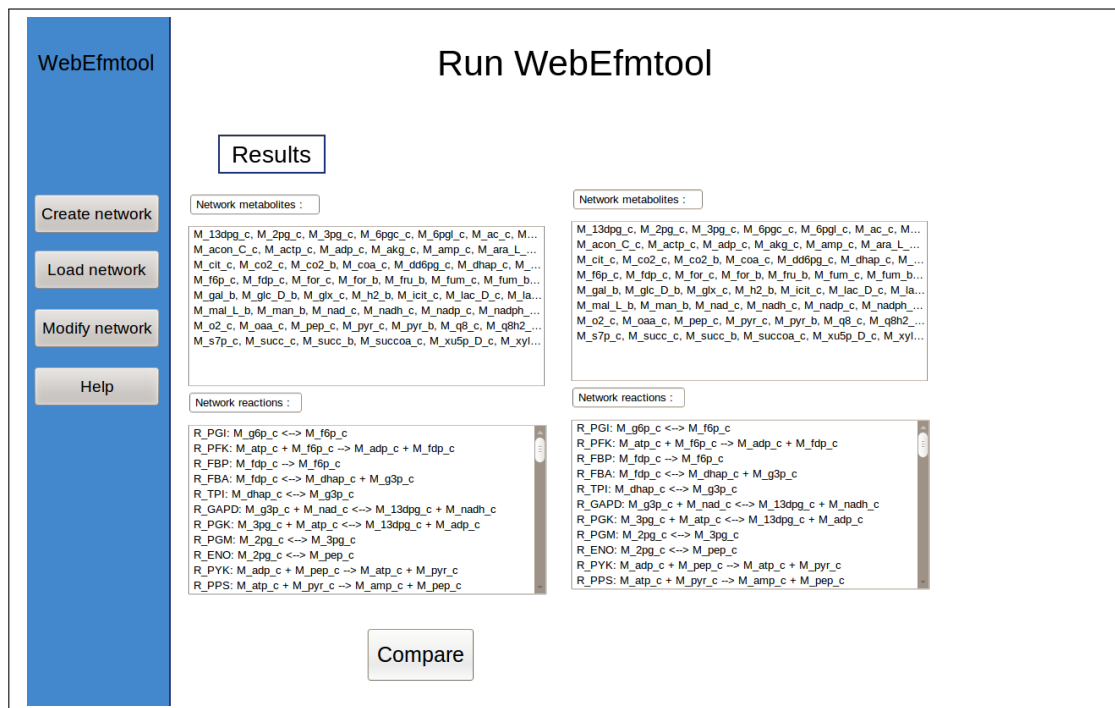


FIGURE 3.7 – Exemple de comparaison des résultats pour *E. Coli*

Pour la comparaison des réseaux, la Figure 3.7 propose un exemple d’affichage en vis-à-vis des résultats.

3.2 Besoins non fonctionnels

3.2.1 Portabilité

L’utilisation de *regEfmtool* 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.

3.2.2 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. L’utilisateur sera donc informé en cas d’erreur lors du chargement d’un fichier non compatible avec *regEfmtool*, contenant des erreurs d’écritures ou lorsque les paramètres entrés ne sont pas en accords avec la fonction ou l’option sélectionnée.

3.2.3 Temps de calcul

Il faudra effectuer une vérification du nombre de métabolites et de réactions afin d’estimer le temps de calcul. Si ce dernier s’avère trop long, l’utilisateur sera prévenu et

devra confirmer le lancement du processus.

3.2.4 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,...).

4.1 Langages

Nous avons fait le choix de langages impératifs orientés Web pour cette interface pour des raisons personnelles et par pertinence quand au projet demandé. De plus, le choix de créer un logiciel Web pour la création de l'interface de *regEfmtree* va nous permettre d'apprendre de nouveaux langages (pour la majorité des personnes composant ce groupe de projet).

Il a été décidé d'utiliser les langages dédiés aux technologies Web, tels que HTML 5¹ pour l'interface graphique couplé aux feuilles de styles CSS², JavaScript et PHP pour le fonctionnement du site Web, ainsi que la persistance des données pour les fonctions d'analyse et de comparaison de réseaux. D'autres langages pourront être utilisés au fur et à mesure de l'avancement s'ils apparaissent être plus adaptés que ceux précédemment cités.

4.2 Accessibilité

Pour permettre un accès facile à notre programme depuis n'importe quel ordinateur, indépendamment du système d'exploitation et de l'endroit, nous avons décidé de développer notre projet sous la forme d'une application Web. En effet, une connexion à la page Web du site suffira pour avoir accès à toute ses fonctionnalités. *RegEfmtree* est installé et exécuté sur une machine *UNIX*, qui est reliée à d'autres machines sur un réseau local (ou sur internet). De ce fait, n'importe quelle autre machine peut avoir accès au logiciel via son navigateur internet.

Cela permet notamment à des machines, même peu puissantes, de pouvoir effectuer des simulations conséquentes si le logiciel est installé sur un serveur ou un ordinateur performant.

Dans le cas d'une installation sur une machine non reliée à un réseau de partage de *regEfmtree* l'exécution est néanmoins possible.

1. HTML : Hypertext Markup Language.

2. Cascading Style Sheets.

Bibliographie

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