



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*

|   |          |
|---|----------|
| <b>Introduction</b>                               | <b>2</b> |
| <b>1 Contexte</b>                                 | <b>3</b> |
| <b>2 État de l'existant</b>                       | <b>4</b> |
| 2.1 Efmtool . . . . .                             | 4        |
| 2.2 <i>RegEfmtool</i> . . . . .                   | 4        |
| 2.3 METATOOL . . . . .                            | 4        |
| 2.4 CellNetAnalyzer . . . . .                     | 5        |
| 2.5 Yana . . . . .                                | 5        |
| 2.6 Acom . . . . .                                | 6        |
| 2.7 JACoMode . . . . .                            | 6        |
| 2.8 Langages . . . . .                            | 6        |
| <b>3 Besoins fonctionnels et non fonctionnels</b> | <b>7</b> |
| 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 . . . . .            | 8        |
| 3.2.1 Portabilité . . . . .                       | 8        |
| 3.2.2 Sécurité et robustesse . . . . .            | 8        |
| 3.2.3 Temps de calcul . . . . .                   | 8        |
| 3.2.4 Documentations . . . . .                    | 8        |
| 3.2.5 Dates . . . . .                             | 8        |
| <b>4 Choix et justifications</b>                  | <b>9</b> |
| 4.1 Langages . . . . .                            | 9        |
| 4.2 Accessibilité . . . . .                       | 9        |
| <b>Bibliographie</b>                              | <b>9</b> |

## *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 donnée 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.

## *Contexte*

Quelques logiciels disponibles internationalement proposent des outils tels que le calcul des modes élémentaires de flux ou la recherche de minimal *cut sets*. Cependant, ils sont soit dépendants de logiciels non libre comme MATLAB, soit ne possèdent pas d'interface utilisateur conviviale.

## 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).

## 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).

## 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. La dernière version, 4.9, est assez performante sur les petits réseaux métaboliques, mais possède de gros problèmes de gestion de mémoire et de rapidité lors de calculs sur de grands réseaux. 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 :

1. la liste des réactions réversibles, ainsi que celle des réactions irréversibles, avec le nom des réactions,
2. la liste des métabolites internes et externes impliqués dans les réactions,
3. les équations réactionnelles se trouvant dans la section.

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 [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 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 [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. YANASquare [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.

## 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.

## 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.

## 2.8 Langages

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

- MOD PERL combiné avec APACHE et CGI 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 HUNCHENTOOT et PARENSCRIPT offre un bon environnement pour le développement web
- JAVA et ses APPLETS JAVA apparaissent également comme un bon choix pour le développement web
- PHP 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.

## *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 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

à voir !

## 3.2 Besoins non fonctionnels

### 3.2.1 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.

### 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 *regEfmtree*, 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 Documentations

L'écriture du code sera constitué de commentaire qui permettrons 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,...).

### 3.2.5 Dates

Début du projet mi-octobre 2012. Remise du cahier des charges et présentation du sujet le 28 novembre 2012. Mi-décembre 2012 : point sur l'avancement du projet. Rendez-vous pour l'avancement du code en janvier 2013 Remise du projet fin février 2013.

## Choix et justifications

### 4.1 Langages

Le choix de langages impératifs orientés web pour cette interface a été effectué pour des raisons personnelles et par pertinence quand au projet demandé. Tout d'abord le choix de logiciels web pour la création de l'interface de *regEfmtree* a été effectué dans le but d'apprendre de nouveaux langages (pour la majorité des personnes composant ce groupe de projet).

### 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 connection 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.

### 4.3 Base de données

Nous allons utiliser la base de données du KEGG afin de permettre une visualisation des réseaux métaboliques. C'est un moyen plus convivial d'observation des résultats qu'une simple liste de réactions et qu'une série de matrices.

## *Bibliographie*

- [1] <http://www.biotec.boku.ac.at/regulatoryelementaryfluxmode.html>.
- [2] <http://www.csb.ethz.ch/tools/efmtool>.
- [3] David E. Ruckerbauer Christian Jungreuthmayer and Jürgen Zanghellini. Utilizing gene regulatory information to speed up the calculation of elementary flux modes.
- [4] Axel von Kamp and Stefan Schuster. Metatool 5.0 : fast and flexible elementary modes analysis.
- [5] <http://www.mpi-magdeburg.mpg.de/projects/cna/cna.html>.
- [6] 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.
- [7] <http://yana.bioapps.biozentrum.uni-wuerzburg.de/>.
- [8] M. Beurton-Aimar S. Pérès, F. Vallée and J.P. Mazat. Acom : A classification method for elementaryfluxmodes based on motif finding.