

Projet AGGP - Dossier d'init

Anthony Tschirhard, Marie Paturel, Marion Poirel, Balthazar Rouberol

1 Introduction

2 Contexte et rappel du problème

2.1 Contexte

L'étude des systèmes biologiques complexes par modélisation sous forme de réseau est un moyen efficace de comprendre le fonctionnement intrinsèque à ces systèmes. En effet, il est possible de relier la structure du système aux fonctions de ses composantes. Par exemple, en étudiant la structure d'un réseau social, on peut comprendre comment l'information se diffuse ainsi que le rôle tenu par les utilisateurs dans son relai.

Les réseaux biologiques sont caractérisés par une architecture "sans-échelle", dans laquelle certaines composantes jouent un rôle plus important que d'autres : les hubs. On citera les exemples de Google, Facebook et Twitter sur le réseau Internet ainsi que l'ATP dans le réseau métabolique de la cellule. Cette architecture implique une distribution des degrés des noeuds selon une loi de puissance de paramètre γ , compris entre 2 et 3 pour la grande majorité des réseaux biologiques ($\gamma_{moy} \simeq 2.1$). Une autre caractéristique d'un tel réseau est la formation de cliques au sein de sa structure.

2.2 Objectifs

Le but de ce projet est de générer un réseau biologique en utilisant un algorithme génétique, développé en séance d'Optimisation. Il faudra donc pour cela décider d'une fonction de fitness adaptée à ce type de réseau, afin que les réseaux générés se rapprochent de génération en génération d'un réseau biologique "idéal".

Les propriétés d'architecture "sans-échelle" et de modularité du réseau sont relativement contradictoires : la présence de hubs rend relativement improbable l'isolation de certains noeuds du réseau, ce qui est pourtant suggéré par les cliques dans contenues dans la structure. Il faudra donc construire la fonction de fitness autour d'un compromis acceptable entre ces deux conditions.

3 Documents de référence

Les connaissances générales sur la structure, l'architecture et le comportement des réseaux biologiques nous viennent de deux articles :

- "Exploring complex networks", *Steven H. Strogatz*, Nature, Vol. 410, March 2001
- "Network biology : understanding the cell's functional organization", *Albert-Lázlo Barabási, Zoltán N. Oltvai*, Nature reviews, Genetics, Volume 5, February 2001

L'algorithme génétique est celui contenu dans le fichier correction `pyAG.py`, fourni par M. Hédi Soula.

4 Contraintes générales

4.1 Contraintes

4.2 Risques

5 Organisation du travail

5.1 Rôles distribués

5.2 Règles de suivi

Le suivi des tâches et des implications de chacun se feront en continu tout au long de ce projet. Au début de chaque session de travail, l'équipe se réunira afin de :

- Contrôler l'avancement du travail de chacun et les éventuels problèmes rencontrés ;
- prendre des mesures correctives en cas de retard dans le projet ;
- faire le bilan sur la partie du projet actuellement abordée ;
- répartir le travail pour la séance.

En fin de chaque session de travail, l'équipe se réunira une nouvelle fois afin de :

- Contrôler l'avancement du travail de chacun et les éventuels problèmes rencontrés ;
- faire de bilan de la séance achevée ;
- répartir de le travail pour la prochaine session ;
- définir une date pour la prochaine séance de travail.

5.3 Organigramme des tâches

5.4 Outils utilisés

Afin de gérer au mieux ce projet, il convenait de mettre toutes les chances de notre côté en utilisant des outils adaptés à ce genre de travaux. Deux outils nous paraissaient particulièrement nécessaires :

- Un outil permettant de partager du code, à la manière de *subversion*, afin que nous puissions tous travailler en collaboration de manière efficace ;
- un outil de gestion de projet, afin que nous puissions tous définir précisément nos objectifs, nos contraintes temporelles et avancées.

Git

Pour le logiciel de gestion de versions, nous avons décidé d'utiliser **Git**, un logiciel libre simple et efficace, dont la principale tâche est de gérer l'évolution du contenu d'une arborescence. Un logiciel de gestion de versions agit sur une arborescence de fichiers afin de conserver toutes les versions des fichiers, ainsi que les différences entre les fichiers.

Ce système nous permettra notamment de mutualiser le développement de notre projet. Nous nous servirons de cet outil pour stocker toute évolution des codes sources ou des rapports – le système conserve en effet une trace de chaque changement. Chacun doit être accompagné d'un commentaire. Le système travaille par fusion de copies locale et distante, et non par écrasement de versions. De cette manière, nous pourrons travailler de concert sur une même source, les changements de l'un n'affectant pas ceux de l'autre.

Pour héberger ce projet, nous avons décidé d'utiliser le site de stockage de projet **GitHub**. Ce service propose l'hébergement de projets sous Git, mais dispose également de fonctionnalités de type réseaux sociaux – suivi de personnes ou de projets, graphes de réseau pour les dépôts etc. Après avoir ouvert des comptes à nos noms et avoir autorisé nos accès par des clés, nous sommes maintenant en mesure de travailler en collaboration de manière efficace et structurée.

Il a été convenu la chose suivante : commencer par récupérer le dépôt disant et le fusionner avant de commencer une nouvelle session de travail, effectuer des *commit* réguliers et bien commentés, et enfin pousser les modifications en fin de session. De cette manière, tout le monde est bien synchronisé avec les autres membres du groupe et le suivi des modifications est propre.

Redmine

Afin de nous organiser et de mettre en forme les différents défis auxquels nous allons devoir faire face, nous avons décidé d'utiliser une application web Open Source de gestion de projet en mode web : **Redmine**. Ses principales fonctionnalités sont la gestion multi-projets, la gestion des développeurs et de leurs rôles, des notifications par mail, des historiques, forums, wiki etc.

Nous allons utiliser cette plateforme pour nous organiser dans notre travail – collectif et individuel –, déterminer les tâches à accomplir, les dates de rendez-vous, de rendus et les progressions des différents membres du groupe.

- ★ Possibilité de voir les différentes tâches à accomplir et les délais impartis ;
- ★ possibilité de consulter les demandes en cours et d'en émettre (en précisant son échéance, sa durée, son statut, à qui elle va être assignée etc.) ;
- ★ évaluation de l'avancée de chacun et visualisation de celle des autres membres du projet.

6 Conclusion