EGN SERVEUR

Auteurs :

ANTOINE DALLAIRE

LOGAN SCHWARTZ

4 mai 2016

Introduction

Les réseaux de similarité sont de plus en plus étudiés en biologie, et plusieurs outils existent pour visualiser les réseaux par exemple Cytoscape ou Gephi. Cependant peu d'outils offrent la possibilité de générer des réseaux de similarité de façon conviviale. EGN, développé par Sébastien Halary, à l'Institut de Recherches en Biologie Végétale, tente d'apporter un peu de convivialité dans la construction de réseaux de similarité génétique. Il permet, ainsi, aux biologistes n'ayant pas beaucoup de connaissances informatiques de construire des réseaux façon interactive. Les réseaux créés par partir d'EGN sont facilement visualisables par Cytoscape ou Gephi. Le but initial de ce projet est de mettre au point une application WEB encapsulant EGN.

Développement

EGN

Le premier défi rencontré était de comprendre et restructurer le programme EGN. Bien que sa nature interactive via un émulateur de terminal le rend simple d'utilisation, elle empêche l'utilisateur de lancer le programme avec des options sans avoir à répondre aux questions. Il est donc difficile d'encapsuler un programme qui se comporte de la sorte et le rendre utilisable via une interface graphique. Le premier défi était évidemment d'apprendre les bases du langage PERL, en effet aucun des deux coéquipiers n'avait eu l'occasion d'utiliser ce langage. Le second, défi était de modifié le code pour qu'il soit possible de passer les options directement lors de l'appel du programme, plutôt que d'avoir à répondre interactivement à chacune des questions. Avec les modifications faites, il est maintenant possible de lancer le logiciel et de faire directement des réseaux. Cependant, à cause d'une hétérogénéité dans la méthode de l'intégration des paramètres, le programme modifié pour être en lancée en ligne de commande ne couvre pas l'ensemble des options disponibles lors de l'utilisation interactive. Par exemple, l'utilisation de BLAT n'est pas disponible, ce qui dans notre cas ne pose

pas trop de problèmes puisqu'il est tout même possible d'utiliser BLAST. De plus, BLAST peut prendre en entrée des fichiers mixtes contenant à la fois des séquences d'acides aminés et de séquence nucléotidiques. Pour remédier au problème de l'hétérogénéité dans la méthode de l'intégration des paramètres d'entrée, nous avons fait un script permettant de produire à partir du formulaire un fichier de configuration pour les paramètres de BLAST alors que les paramètres de l'analyse de réseau sont passés en ligne de commande.

EGN web

L'idée du projet repose sur l'implémentation d'une interface web 2.0, où l'utilisateur n'aurait qu'à téléverser un fichier de séquences fasta, choisir les paramètres voulus et par un simple clique un réseau de similarité serait générer et pourrait être visualisé à même le navigateur web. De plus, le graphe de similarité devait avoir un minimum d'interactivité et d'informations. Cette redirection du projet avait le double avantage de nous permettre de réutiliser le travail fait pour lancé EGN en ligne de commandes, et d'utiliser des technologies web que nous avions déjà apprises dans le cadre d'autres cours.

Choix technologiques

Pour avoir un serveur web avec une interface graphique fonctionnelle en moins d'une semaine, nous avons choisir des technologies qui nous étaient familières. Pour le serveur, nous avons utilisé *flask* , un cadriciel minimaliste développé en python et qui est facilement extensible, la majorité du code coté-serveur est donc du python. Pour le côté client, *Bootstrap* a été utilisé pour l'aspect esthétique de l'interface, *Angular.js* pour l'interaction client-serveur. Dans le cadre de la présente application, nous n'avons pas utilisé les capacités de lier les données entre elles (‘*data binding*’), mais elle permettrait d'ajouter facilement d'autres fonctionnalités dans le futur permettant entre autres de visualiser des informations complémentaires ou supplémentaires qui seraient synchronisées avec les cliques de l'utilisateur dans le réseau. Enfin pour la visualisation du réseau notre choix s'est

arrêté sur *sigma.js* un outil simple à apprendre, mais permettant de faire de bonnes visualisations.

Flux d'une analyse

Voici maintenant une description générale du fonctionnement du serveur WEB. Tout d'abord, l'utilisateur ouvre un navigateur et entre l'adresse de EGN server, dans le cas présent le serveur tourne localement et écoute sur le port 8084 l'adresse est donc http://127.0.0.1:8084. Une version en ligne est accessible temporairement à l'adresse http://binfo08.iric.ca:8084 . On voit alors un page très simple avec un réseau d'exemple et deux boutons en haut à droite (Figure 1).

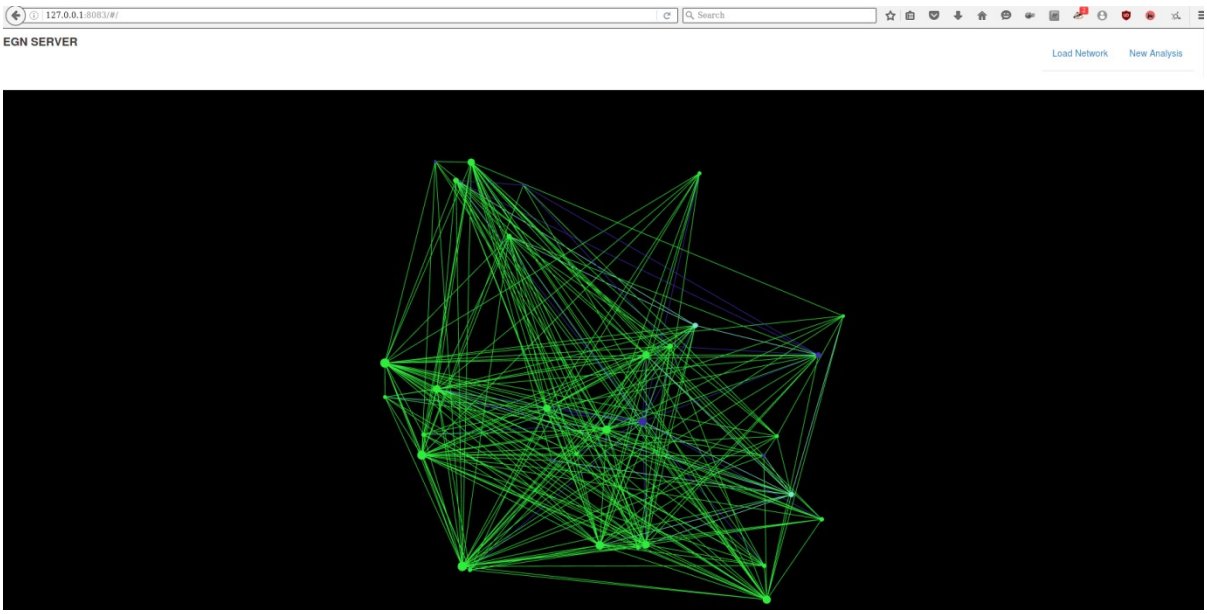

Figure 1. Page d'accueil d'EGN server, avec réseaux de démonstration.

Le bouton *New Analysis*, comme son nom l'indique, permet de faire une nouvelle analyse. En cliquant dessus, une fenêtre apparaît. Cette fenêtre indique à l'utilisateur d'entrer les différents paramètres à considérer lors de l'analyse (Figure 2). On peut voir les paramètres par défaut pour BLAST ainsi que pour la construction du réseau de similarité. Si l'utilisateur désire modifier les paramètres, c'est ici qu'il doit le faire. Aussi l'utilisateur doit inscrire son adresse courriel il pourra

ainsi recevoir les résultats à cette adresse. De plus, il doit choisir un fichier fasta d'acide aminé ou d'acide nucléique dans son ordinateur il peut le faire en cliquant sur *Browse* ou *parcourir* et ensuite parcourir les répertoires de son ordinateur à la recherche du fichier voulu.

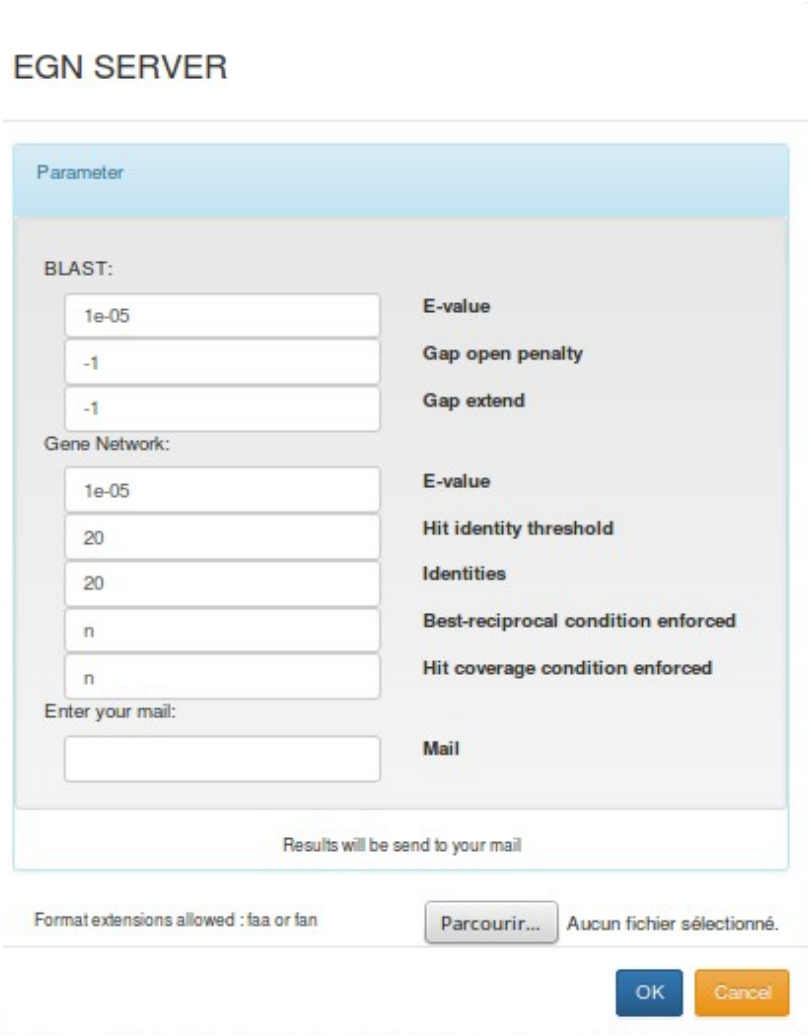

Figure 2. Fenêtre de configuration de nouvelle analyse

Deux mécanismes vérifient que le fichier sélectionné est valide, la première vérifie que l'extension du fichier. Le second mécanisme est plutôt un avertissement donné lors du lancement d'EGN avec le fichier, si l'extension est bonne, mais que le fichier n'est pas du bon type, un autre message d'erreur est alors affiché (Figure 3). Le second message d'erreur s'affiche aussi si l'utilisateur choisit un seuil de similarité

tellement haut que EGN n'est pas capable de créer de liens entre les noeuds. Si tous les paramètres sont valides, l'analyse est alors lancée. Tout d'abord un BLAST est fait sur l'ensemble des séquences, ensuite en fonction des résultats du BLAST, un réseau de similarité est construit selon les paramètres préalablement établis.

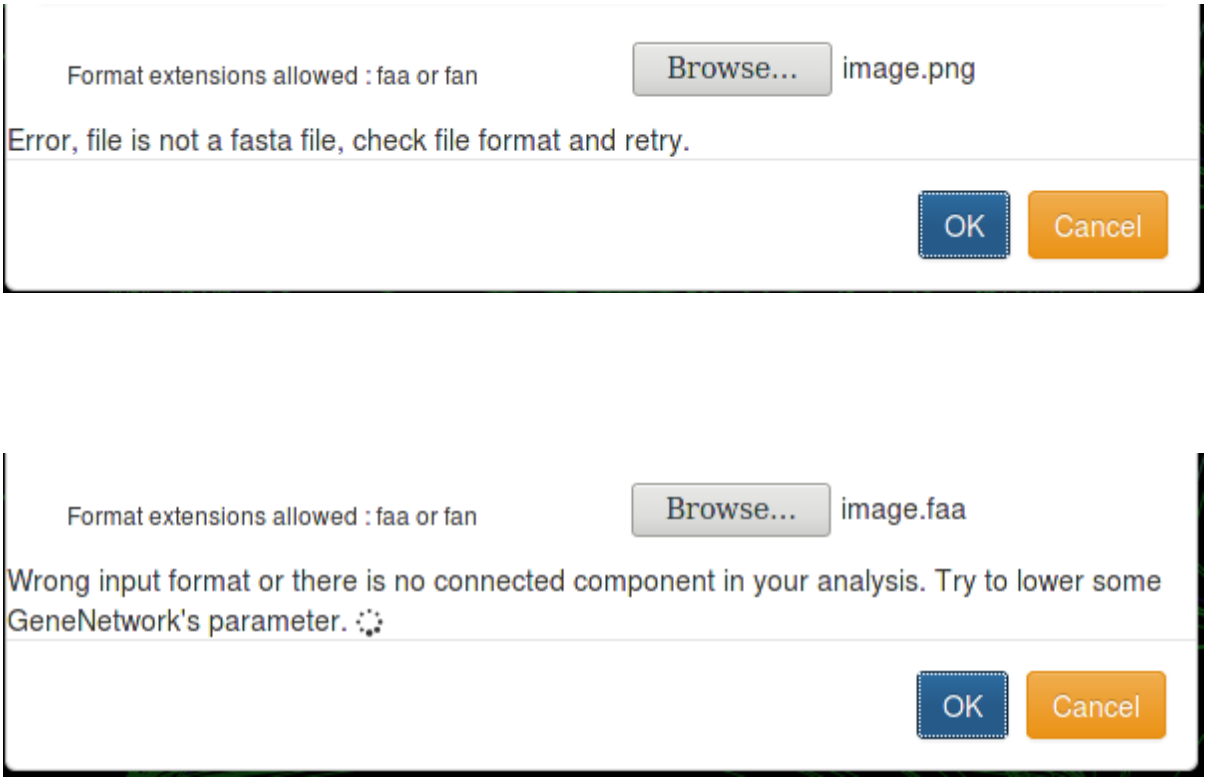

Figure 3. Contrôle du fichier d'entrée par vérification de l'extension(en haut), et type de fichier (en bas).

D'un point de vue plus technique, lorsque l'on téléverse le fichier à analyser, le serveur crée un répertoire portant le nom du fichier d'entrée dans le dossier *workflow*. Un fichier de configuration est aussi généré, à partir des choix faire par l'utilisateur, et est inséré dans le répertoire. Ensuite, le serveur copie EGN dans le même répertoire, car, de par sa conception, EGN ne s'exécute que sur des fichiers du répertoire dans lequel il se trouve, c'est un défaut auquel nous n'avons pas trouvé de solution simple. Si jamais l'utilisateur veut faire des analyses avec des paramètres différents, mais sur un même fichier d'entrée, toutes les analyses

associées à ce fichier d'entrée seront ajoutées par la suite dans le même répertoire dans des dossiers différents indiquant les paramètres utilisés pour l'analyse (ex: GENENET_1e-05.100.20.0.0). Ainsi une fois une multitude d'analyses faites, le tout peut être envoyée à l'utilisateur au format ZIP.

L'autre bouton, *Load network,* permet à l'utilisateur de charger de réseaux au format .gexf ou .json directement, sans passer par les étapes de création du réseau d'EGN. Le serveur prend alors le fichier, le sauvegarde dans le dossier *uploads*. Il changera aussi son nom pour *network*, tout en gardant l'extension d'origine. Ensuite, il sera déplacé dans le répertoire *Front.* En fonction de l'extension, l'utilisateur sera redirigé vers la page appropriée de visualisation du réseau avec *sigma.js*.

Comme mentionné précédemment, EGN génère des réseaux visualisables dans Cytoscape et dans Gephi. Cependant ces deux logiciels ne prennent pas le même genre de fichier entré. Pour Gephi, EGN génère un fichier XML ayant comme extension .gexf. Le format du réseau de similarité généré pour Cytoscape consiste en 2 fichiers, l'un avec le nom et l'identifiant de chacun des noeuds l'autre avec les différentes arêtes du réseau ainsi que des statistiques associées à chacune des arêtes (Figure 4). Ces fichiers sont envoyés par courriel à l'adresse fournie, et pourront par la suite être visualisés à l'interface web, ou avec un autre logiciel au choix de l'utilisateur.

ID	SAMPLE
5	Archaea
29	Bacteria
17	Bacteria
7	Eukarya
16	Bacteria
24	Bacteria

ID1	ID2	HIT%ID	SS%ID	EVALUE	MIN_COV_OPT
17	29	62.57	62.21	69	1
5	29	47.37	47.09	48	1
5	17	45.61	45.34	49	1
16	5	45.61	45.34	44	1
16	29	62.57	62.21	67	1
16	17	63.74	63.37	67	1

Figure 4. fichier de noms et identifiant (à gauche), fichier d'arêtes et statistiques (à droite).

Visualisation

Pour faire un rendu graphique du réseau dans le navigateur web nous avons choisi *sigma.js*, il est possible, avec ce cadriciel, d'uniquement fournir le fichier .gexf et d'afficher le réseau. Cependant, nous avons remarqué les fichiers générés pour Cytoscape contenait plus d'informations pertinentes pour l'utilisateur et nous voulions nous assurer qu'il puisse avoir accès à ces informations dans la représentation graphique. Malheureusement, *sigma.js* ne permet pas d'afficher un réseau à partir des 2 fichiers pour Cytoscape. Il était donc nécessaire de convertir ces deux fichiers en un seul fichier au format JSON pour organiser les informations de manière à ce que *sigma.js* puissent les interpréter. Pour ce faire, un script python prenant en paramètre les 2 fichiers et créant une sortie JSON a été conçu. Afin de représenté la e-value de chacune des arêtes et ainsi avoir une information supplémentaire sur le réseau, le script génère un gradient de couleur relatif pour chacune des *e-values*. Il est alors possible de voir en un coup d'oeil, la *e-value* entre 2 noeuds. Ce script est facilement modifiable afin d'ajouter de nouvelles clés au JSON qui pourront par la suite être visualisable dans le réseau. Un autre point important concernant le fichier du JSON est que les positions x et y de chaque noeud sont générées aléatoirement, puisque cette information n'était pas présente dans nos les sorties de EGN. Idéalement, il faudrait ajuster la position afin de diminuer la distance des arrêtés entre les noeuds ayant un seuil de similarité plus grand, ainsi la visualisation contiendrait encore plus d'informations. Un autre avantage de notre script est que le JSON généré permet d'afficher l'ensemble de l'information contenue dans un seul réseau. Alors que, dans certains cas, l'analyse produisait de multiples sorties Gephi, ce qui rendait la visualisation complète de l'information plus difficile.

Interaction utilisateur

Dans le réseau produit par *sigma.js*, la taille d'un noeud est dépendante du nombre de liens auquel il est attaché, la couleur des noeuds quant à elle est générée en fonction du nom des noeuds ou du groupe auquel il appartient dépendamment du

format du réseau. Si l'utilisateur utilise le format JSON, il pourra reconnaître les noeuds appartenant au même groupe (Figure 5, image de gauche). Dans l'exemple utilisé pour ce rapport il y a trois noms de noeuds, Archaea, Bacteria et Eukarya auxquels sont attribuées, respectivement, les couleurs, violet, vert et bleu. Lorsque l'utilisateur clique sur un noeud du réseau, tous les noeuds et liens qui y sont rattachés sont colorés en rouge alors que les autres sont mis en noir (Figure 5, image du centre). Un clique droite sur ce même noeud permet alors de visualiser la *e-value* des arrêtes partagées avec ses noeuds voisins selon un dégradé allant du rouge au bleu (du plus faible au plus grand), les autres noeuds et arrêtes du réseau sont colorées en blanc (Figure 5, image de droite). Un double clique sur l'arrière-plan permet de revenir à la visualisation du réseau par groupe/nom. *Sigma.js* implémente par défaut certain fonctionnalités tel que le zoom avec la roulette de la souris, la possibilité de se déplacer dans le réseau en maintenant le clique pendant que l'on bouge la souris où encore un 'tooltip' qui présente le nom du noeud lorsqu'on le survol.

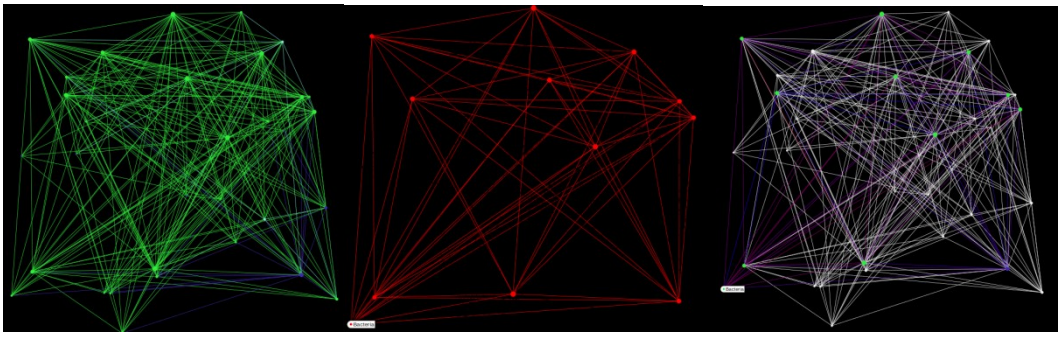

Figure 5. À gauche est représenté un réseau entier. Au centre seulement sont représentés les noeuds voisins du noeud qui vient d'être sélectionné par l'utilisateur et les liens les reliant. À droite, les mêmes liens sont colorés en fonction de leur e-value, et les autres noeuds et liens du réseau sont colorés en blanc.

Conclusion

Pour conclure, EGN est un outil bio-informatique très intéressant, mais qui souffrait d'un manque de flexibilité concernant l'utilisation par ligne de commande. L'interface graphique que nous avons faite élimine ce problème. Elle permet, entre autres, de lancer plusieurs analyses, ayant des paramètres différents, de façon successive sur un même fichier d'entrée afin de récupérer l'ensemble des résultats fournis par EGN pour toutes les analyses par courriel dans un fichier au format zip.

De plus, les scripts que nous avons faits pour transformer les sorties destinées à cytoscape en JSON, permettent d'ajouter des fonctionnalités inexistantes, à notre connaissance, au niveau de la visualisation. Étant donné les dernières modifications du code de notre application, l'analyse des réseaux de génome serait également possible sans trop de modifications.

L'implémentation d'*Angular.JS* du coté client créer une base laissant la possibilité d'ajouter des fonctionnalités permettant de voir l'ensemble de l'information associé à un noeud de façon plus interactive et agréable. On pourrait par exemple utiliser plusieurs outputs de réseaux (avec des seuils différents) pour afficher l'évolution du réseau en fonction du seuil d'identité ou d'un autre paramètre.

Finalement, notre interface apporte une base laissant libre champs à l'implémentation de nouvelles fonctionnalités qui pourraient faire de ce projet, un outil bioinformatique à part entière s'il était poussé plus loin.