Master 2 bioinformatique
Projet Python (1)

19 septembre 2025

Version 1.1

1 Préambule : Contexte Biologique

La recherche exploratoire se caractérise par l'absence d'hypothèse que l'on cherche à valider. Au contraire, la recherche exploratoire a pour but d'identifier de nouvelles potentielles hypothèses. C'est par exemple ce que l'on fait fréquemment lorsqu'on réalise des analyses -omiques à haut débit entre une condition contrôle et une condition que l'on cherche à caractériser. Avec ces analyses -omiques haut débit, on va (i) mettre en évidence tout un ensemble de gènes ou protéines liés à la condition qu'on cherche à caractériser ; (ii) analyser les processus biologiques liés à cet ensemble de gènes/protéines ; (iii) on en déduira l'hypothèse que les processus biologiques identifiés sont clés dans la condition que l'on cherche à caractériser.
Prenons l'exemple d'une analyse transcriptomique (type NGS), qui compare le transcriptome du tissu adipeux chez des personnes non malades et chez des personnes souffrant d'un syndrome métabolique. L'analyse des données nous permettra d'identifier des gènes différentiellement exprimés (item i). Ensuite, on pourra regarder les annotations de ces gènes/protéines pour identifier dans quel processus cellulaire X,Y,Z ils/elles interviennent (item ii). Cela permettra de formuler les hypothèses H_1 que le processus X est dérégulé chez des personnes souffrant de syndrome métabolique, H_2 que c'est le processus Y, H_3 que c'est le processus Z, H_4 que ce sont les processus X, Y et Z qui sont simultanément dérégulés, etc. Chacune de ces hypothèses peut ensuite être abordée par la méthode expérimentale classique afin d'être infirmée ou non [1].

Dans ce sujet, on s'intéresse plus particulièrement à l'étape (ii), qui consiste, à partir d'une ensemble de gènes/protéines, à identifier quels sont les processus cellulaires clés reliés. Deux grandes familles de méthodes existent pour faire cela.

— Les méthodes ORA pour *Over Representation Analysis*, qui se basent sur la fréquence d'une annotation dans notre ensemble, comparée à la fréquence si on prenait au hasard un groupe de protéines [2]. Par exemple, si l'annotation `DNA damage checkpoint` est retrouvée pour 25% des gènes/protéines du groupe, alors qu'au hasard on s'attend à trouver autour de 10%, il

1. contrairement aux mathématiques, en sciences expérimentales on ne peut jamais montrer qu'une théorie est vraie

2. un groupe au hasard ayant exactement la même taille, car cela influe sur le résultat et notamment sur le calcul de la p-value

y a sur-représentation de l'annotation `DNA damage checkpoint`. D'un point de vue statistique, les calculs de sur-représentations sont généralement basés sur la loi hypergéométrique. Comme on effectue un test statistique pour chaque annotation, et qu'il y a en général plusieurs milliers ou dizaines de milliers d'annotations, il faut systématiquement faire une correction pour les tests multiples. De très nombreux outils bioinformatiques existent pour calculer ces enrichissements.

— D'autres méthodes sont basées sur le principe GSEA pour *Gene-Set Enrichment Analysis*. Dans une analyse de type GSEA, on va classer tout les gènes, qu'ils soient différentiels ou non, dans l'ordre du plus sur-exprimé au plus sous-exprimé (ie. du fold-change le plus grand au plus petit). Ensuite, pour une annotation donnée, on va regarder si les gènes/protéines qui portent cette annotation sont distribués uniformément dans la liste triée (l'annotation n'est pas pertinente), ou au contraire si la distribution n'est pas uniforme. Dans ce dernier cas, on peut observer que les gènes impliqués dans le processus de l'annotation sont plutôt tous sur-exprimés, ou plutôt tous sous-exprimés, ou encore tous fortement affectés si le processus fait intervenir des gènes activés et d'autres gènes inhibés. De très nombreux outils bio-informatiques utilisent cette technique, et on la dit plus robuste car on analyse tous les gènes et pas seulement les gènes différentiellement exprimés : ainsi, il n'y a pas d'effet seuil autour du seuil de 5% utilisé classiquement pour dire si un gène est différentiellement exprimé ou non.

Ces méthodes existent, et l'objet de ce projet n'est pas de vous demander de les ré-implémenter. Cela dit, on va vous demander d'en implémenter une autre :-)

Les deux approches listées ci-dessus présentent cependant un double problème : elles renvoient généralement beaucoup (trop) d'annotations pertinentes, et il y a une certaine redondance entre certaines des annotations pertinentes. Cela vient du fait que l'on utilise des annotations avec différents degrés de précision : soit parce qu'au moment de générer l'annotation on n'a pas toutes les informations (par exemple, pour décrire une photo avec des oiseaux on utilise l'annotation « oiseau » si on ne connaît pas le nom de l'espèce ou que ce n'est pas pertinent), soit parce que l'on souhaite faire des requêtes plus ou moins précises (par exemple, « retrouver toutes les photos d'hirondelles » aussi bien que « retrouver toutes les photos d'oiseaux »). Il existe des dépendances entre certaines annotations (si on annote une photo avec « hirondelle », on doit également l'annoter avec « oiseau » et avec « animal »). Ces dépendances sont typiquement formalisés dans des ontologies.

Les ontologies sont une très bonne structure de données pour représenter les connaissances que l'on a sur la fonction des gènes[3], car elles permettent de hiérarchiser les annotations et faire des liens entre elles. Par exemple, si une protéine est annotée "positive regulation of transcription", alors, même si ce n'est pas écrit, elle est aussi annotée "regulation of transcription", parce que "positive regulation of transcription" est un enfant de "regulation of transcription". Les ontologies permettent ainsi à la fois de simplifier le processus d'annotation : on annote aussi précisément que possible et toutes les annotations plus générales viennent automatiquement (en remontant dans la hiérarchie, donc), et d'enrichir le processus de requêtage : lorsqu'on fait référence à une annotation, on cherche également tous les éléments qui sont annotés par une annotation plus précise (cette fois en descendant dans la hiérarchie). Par exemple, si une protéine est annotée "positive regulation of transcription", alors, même si ce n'est pas écrit, elle est aussi annotée "regulation

3. et sur les connaissances en biologie en général, vous pourrez discuter de cela dans l'UE DEL

of transcription", parce que "positive regulation of transcription" est un enfant de "regulation of transcription". Inversement, une requête sur les protéines annotées par "regulation of transcription" devrait également renvoyer les protéines annotées par "positive regulation of transcription".
Cette organisation hierarchisée des connaissances est très souple et permet de représenter la connaissance avec la granularité dont on dispose (on ne sait pas toujours si la régulation est positive ou négative, donc c'est utile de pouvoir dire "régulation positive", "régulation négative" ou juste "régulation"...), mais elle permet aussi de faire des ponts entre différents processus qui participent à une même grande fonction. Par contre, cette organisation des connaissances induit quelques désagréments : (i) il y a beaucoup d'annotations ; (ii) quand on ajoute tous les ancêtres aux annotations d'une protéine, les protéines finissent par avoir énormément d'annotations ; (iii) on ajoutera, pour les plus matheux, que statistiquement ces annotations ne sont pas indépendantes les unes des autres.

La plus célèbre ontologie qui annote la fonction des gènes est connue sous le nom de `Gene Ontology` (acronyme GO) : vous pouvez la visiter le site `https://geneontology.org/`. Dès la page d'accueil, on vous proposera de faire une analyse d'enrichissement :-) Allez explorer la page d'accueil, regardez un peu les onglets proposés, et essayez par exemple de répondre à la question suivante : quels sont tous les ancêtres (ou termes parents) du terme "G2/M transition of mitotic cell cycle". [4].

2 Objectif

Caractériser un ensemble de protéines en fonction des annotations des protéines qui le composent est un problème difficile, sur lequel la communauté scientifique travaille depuis au moins deux décennies et qui demeure une question ouverte.

Dans ce sujet, on vous demande de tester une nouvelle méthode à l'aide d'un algorithme auquel nous avons pensé, mais que nous n'avons encore jamais testé. Cet algorithme a la particularité de se baser uniquement sur 3 notions :

1. le contenu d'information des annotations (IC pour *Information Content*) : dans l'idée, on aimerait bien que les processus cellulaires clés reliés à notre ensemble de gènes soient assez précis (ie. et ne soient pas trop vagues).
2. la couverture : dans l'idée, on aimerait que les processus cellulaires clés que l'on choisit représentent une grande partie des gènes/protéines de l'ensemble.
3. les relations parents - enfants entre les annotations : dans l'idée, une fois qu'on a dit "response to stimulus", cela n'a pas vraiment de sens de sélectionner ensuite "positive response to stimulus", car cet aspect était déjà traité par l'annotation précédente.

Vous êtes donc dans une situation de projet typique : nous n'avons pas tout testé au préalable, et nous allons devoir travailler ensemble pour aboutir à une solution. Soyez force de proposition, et soyez prompts à nous remonter vos points de blocage !

Comme c'est une question scientifique ouverte et une nouvelle méthode que nous vous demandons d'implémenter, on attend de vous :

4. Au cas où, son identifiant est GO :0000086

— que vous vérifiez bien la correction de votre méthode : est-ce que la méthode respecte les spécifications qu'on vous a données ;
— que vous évaluiez vos résultats en les comparant aux solutions existantes.

Entrée Votre programme prendra en entrée une liste d'identifiants UniProt nommée L (comme les exemples sur moodle). Vous aurez aussi besoin de divers fichiers décrits au chapître 5.

Sortie Ce que l'on attend en sortie c'est une petite liste d'annotations (ie. de termes Gene Ontology) qui caractérise l'ensemble de la liste L passée en entrée.
Comme dit dans l'énoncé, les ensembles de termes GO renvoyés par les méthodes classiques ont tendance à être encore assez verbeux, et surtout comportent de la redondance (on retrouve dedans des annotations qui se ressemblent). On aimerait donc récupérer :
— a minima un ensemble de termes GO si possible pas trop gros et pas trop redondant
— éventuellement chaque terme GO peut être agrémenté de scores si vous en calculez
— si vous observez qu'il existe une structure entre certains de ces termes GO, vous pouvez aussi la faire apparaître.

3 Déroulement

La méthode à laquelle nous avons pensé, basée uniquement sur le contenu d'information, la couverture, et la notion de parent-enfant dans les annotations est décrite brièvement dans le document PDF ci-joint. Prenez-en connaissance.

La première chose que nous attendons de vous est que vous nous remontiez ce dont vous avez besoin pour mener votre projet à bien. Nous savons déjà que vous aurez certainement besoin de la librairie python `goatools`, pour laquelle vous trouverez de la documentation ici : `https://github.com/tanghaibao/goatools`. Pour le reste, ne bloquez pas si il vous manque quelque chose et que vous nous avez envoyé un email : si vous ne pouvez pas travailler sur une partie, travaillez sur une autre le temps que nous vous répondions. C'est une bonne habitude à prendre et cela vous forcera à séquencer votre projet (ie. le découper en work packages assez autonomes).

Pour nous contacter, merci d'envoyer un email systématiquement à :
— `emmanuelle.becker@univ-rennes.fr` et
— `olivier.dameron@univ-rennes.fr` et
— `marine.jacquier@univ-rennes.fr`.
Nous vous répondrons dès que possible. Dans tous vos emails, vous préciserez dans l'objet `M2 BI - projet annotations`, et vous indiquerez quels sont les membres du binôme.

4 Calendrier

Le rendu du projet est un code + un rapport (environ 4 pages figures incluses) qui devra :
— présenter votre projet et comment vous avez structuré le code

— présenter (et parfois justifier) les choix techniques que vous aurez faits
— que la méthode fait ce qu'on vous a demandé (validation)
— comparer la méthodes aux autres méthodes (évaluation).

Vous devrez transmettre votre rapport à deux personnes du groupe classe (désignées ultérieurement) au plus tard le 2 octobre. Les deux personnes devront vous faire un retour sur votre rapport pour le 6 octobre (en 1 page max par personne). Vous aurez ensuite jusqu'au 8 octobre pour prendre en compte les retours (ou pas si vous ne les jugez pas pertinents), et soumettre les deux évaluations par vos camarades que vous avez reçues + votre rapport final.

Vous serez notés à la fois sur la base de votre rapport, mais aussi sur la base de ce que vous avez fait comme retour sur le travail des autres.

5 Données disponibles

5.1 GO : Hiérarchie de GeneOntology

La hiérarchie de GeneOntology est composée de 3 branches, de taille et de structures très inégales :

— **BP : *Biological process*** (`GO:0008150`) décrit l'aspect fonctionnel des processus biologiques dans lesquels la protéine intervient ;
— **CC : *Cellular component*** (`GO:0005575`) décrit les compartiments cellulaires où la protéine a été observée ;
— **MF : *Molecular function*** (`GO:0003674`) décrit le mode d'action biochimique de la protéine à l'échelle moléculaire.

Dans le cadre de ce projet, vous vous focaliserez sur la branche « *Biological process* » (BP) de GeneOntology. La structure de Gene Ontology est représentée dans différents formats (menu « Downloads » puis sous-menu « Download ontology » du site web), qui seront abordés ultérieurement dans l'UE « data engineering in life sciences » (DELiS). Pour ce projet, vous trouverez sur Moodle :

— un fichier `geneOntology-BP-label.csv` qui contient le terme associé à chaque annotation GO de la branche BP (par exemple, « macrophage homeostasis » pour `GO:0061519`) ;
— un fichier `geneOntology-BP-hierarchy-direct.csv` qui contient les couples (sous-classe, super-classe) où la sous-classe est l'identifiant d'une annotation précise, et super-classe d'une annotation plus générale. Dans ce fichier il s'agit de relations directes.
— un fichier `geneOntology-BP-hierarchy-indirect.csv` qui contient les couples (sous-classe, super-classe) où la sous-classe est l'identifiant d'une annotation précise, et super-classe d'une annotation plus générale. Dans ce fichier il s'agit de l'union des relations directes et des relations indirectes, c'est-à-dire que chaque sous-classe est associée à tous ses ancêtres (ou chaque super-classe à tous ses descendants).

5.2 GOA : Annotations

Les descriptions des protéines par des annotations de Gene Ontology sont disponibles au format GAF [5], qui permet de représenter les informations relatives à une association (protéine, classe de GeneOntology), comme par exemple l'evidence code, la référence vers l'article ou la base de données dont provient cette annotation, ou encore divers modifiers. Vous remarquerez qu'il s'agit là des annotations directes des protéines. À vous d'exploiter la hiérarchie de Gene Ontology pour inférer les annotations indirectes. Vous trouverez la version à jour des annotations des protéines humaines dans le fichier `goa_human.gaf` [6]

Dans le cadre de ce projet, vous vous contenterez des informations des colonnes :

- `DB Object ID` (colonne 2) qui contient l'identifiant UniProt des protéines
- `DB Object Symbol` (colonne 3) qui contient le symbole usuel de cette protéine
- `GO ID` (colonne 5) qui contient l'identifiant de l'annotation de Gene Ontology
- `Aspect` (colonne 9) qui indique la branche à laquelle appartient l'annotation GO (filtrez sur `'P'` pour les processus biologiques)
- `DB Object Name` (colonne 10) qui contient la version longue du nom de la protéines de la colonne 3
- `DB Object Type` (colonne 12) filtrez sur `'protein'`

5. https://geneontology.org/docs/go-annotation-file-gaf-format-2.2/
6. http://current.geneontology.org/products/pages/downloads.html