Projet 7: epigenetics marks and targets

Alex Ye

November 10, 2019

1 Introduction

Ce projet se situe durant la phase de la réplication cellulaire où l'ADN est dans son état chromatine (ADN compaqueté autour de histones). Dans cet état là, le génome est susceptible à la modification notamment avec des variantes de histones dont les propriétés physiques vont faciliter ou au contraire compliquer la lecture de certaines sections du génome et donc réguler l'expression génétique. Dans la régulation, on distingue mark et target. Les marks épigénétiques sont des protéines régulateurs qui influencent l'expression génétique, elles peuvent s'activer peu après la réplication cellulaire ou après plusieurs réplications, elles ne se perdent pas après la réplication. Les targets, comme leur nom l'indique, sont les cibles des marks, ce sont les gènes dont l'expression est régulée par une ou plusieurs marks.

Dans ce projet nous allons générer un réseau de régulation d'expression génétique avec les données d'expression génétique livrées avec le projet. Nous utiliserons les algorithmes vu en Cours/TP pour générer des réseaux. Enfin nous mesurerons la performance des réseaux par l'enhancement folding, méthode basée sur des résultats expérimentales qui combine erreur de type I et II.
De ce fait il faudra explorer des bases de donnée pour récolter des données expérimentales d'intéraction entre facteur de transcription.

2 Data

Nos données d'expression génétique sont composées de 26893 observations sur 32 facteurs de transcription (TF).
Afin de receuillir des données expérimentales nous explorons les base de donnée suivantes:

- BIOGRID: 508019 interactions (240 entrées liées à nos TF)
- HI-union: 64006 interactions (0 entrée liée à nos TF)
- HuRI: 52570 interactions (0 entrée liée à nos TF)

Nous comptons principalement sur BIOGRID qui est cité dans l'article ChromNet, cependant BIOGRID ne comporte pas les interactions entre modification de Histone, d'après le même article ces données sont disponible sur le site du projet Histome, mais il semble que le projet soit discontinué. Les liens de téléchargement mènent à des pages vides. Nous essayons de compléter avec d'autres bases de donnée disponibles en ligne telles que HuRI mais elles n'ajoutent aucune informations supplémentaire. Tant pis nous devrons nous en passer.

Des 240 entrées trouvées en tout, nous formons un réseau de 68 arcs dont 12 noeuds reliés à soi-même. Cependant aucune entrée sur les TF: **H3K27ac, H3K27me3, H3K36me3, H3K4me1, H3K4me2, H3K4me3, H3K79me2, H3K9ac, H3K9me1, H3K9me3, H4K20me1, POL2b, RNAPIIS5P**

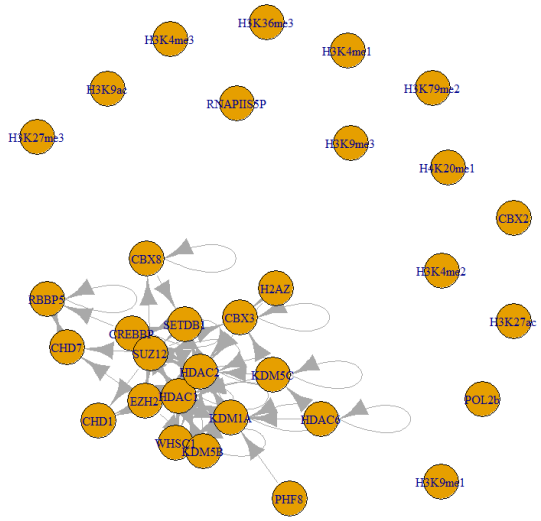

Figure 1: Réseau formé à partir des données expérimentales

Ce réseau servira de référence appuyé par des données expérimentales, il nous permettra de mesurer la qualité des réseaux que l'on va générer par la suite.

3 Methodology

Dans notre cas vu qu'il s'agit d'un réseau de régulation, il nous faut nécessairement un graphe causal afin de savoir quelle mark régule quel target.
Notre choix se porte donc sur les algorithmes:

- **HC**: Devine le réseau sous-jacent "le plus probable" à partir des données (scores bayésiens).
- **PC**: Construit un squelette (réseau sans direction) avec des tests d'indépendence conditionnel puis orienter le réseau en utilisant la structure en V.
- **miic**: Basée sur PC à la différence qu'il prend aussi en compte les théories de l'information et que la densité du réseau dépend entièrement des données.
- **MMHC**: Construit un squelette (réseau sans direction) avec des tests d'indépendence conditionnel puis oriente le réseau en utilisant les scores bayésiens.
- **GENIE3**: Algoritme conçu spécifiquement pour la reconstruction de réseau de régulation, donc à priori le plus intéressant pour nous.
  Décompose le problème en n problèmes de régression pour n gènes. Pour chaque problème de régression, le gène est traité comme régulé par tous les autres en suivant leur expression, formant des arbres de décisions. Ensuite trouver des patterns entre l'expression de chacuns des gènes en utilisant des algorithmes d'arbres. Ne construit pas un réseau à proprement parlé, mais une liste d'interactions classées des plus probables aux moins probables, ces interactions (arcs) peuvent ensuite se transformer en un réseau.

Pour éviter la déviation du à l'initialisation, pour chaque algorithmes nous créerons 5 réseaux avec les mêmes paramètres et moyennerons les scores.

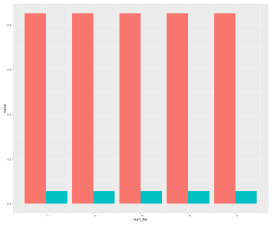

Figure 2: hc

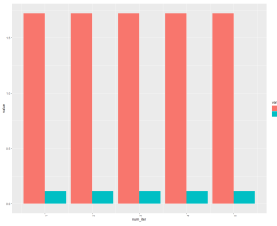

Figure 3: miic

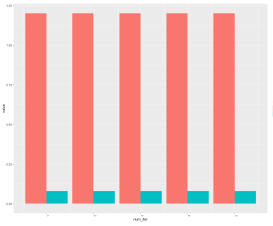

Figure 4: GENIE3

Figure 5: fold enrichment(rouge) et précision(bleu) par iteration de l'algorithme avec les mêmes paramètres

Nous obtenons exactement les mêmes scores à chaque itération même si le plot diffère. Il n'y a en réalité aucune déviation entre 2 appels de la même fonction

de génération de réseau.
Nous pouvons donc optimiser nos paramètres en générant seulement une fois le réseau puisqu'il n'y a pas d'aléa dans l'initialisation. Ce qui signifie aussi qu'il n'y a pas besoin d'optimiser pour les méthodes sans paramètres significatifs tels que HC ou miic.

Figure 6: pc, α

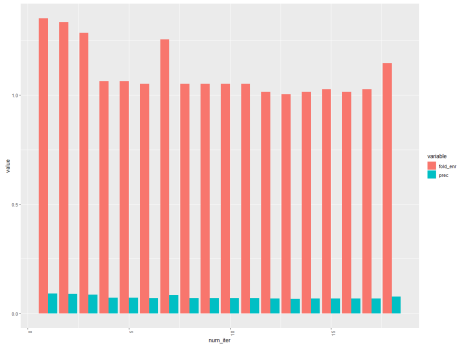

Figure 7: mmhc, α

Figure 8: fold enrichment(rouge) et précision(bleu) par incrément de 0.05 sur α

Le α optimal pour la méthode pc est à 0.4, fold enrichment: 2.039
Le α optimal pour la méthode mmhc est à 0.05, fold enrichment: 1.3514

Dans ce graphique on voit que le fold enrichment est bien proportionnel à la précision, rendant le second inutile et ironiquement moins précis puisqu'il ne prend en compte que si l'arc prédit est dans le réseau de référence ou non, sachant que ce réseau de référence n'est pas forcément complet (dans notre cas on sait qu'il manque les interactions entre histones, soulignant d'autant plus le problème).

Il est cependant intéressant de noter que les scores sont erratiques. Ces algorithmes de génération ayant tendance à prédire les arcs pour lesquels ils trouvent le plus haut indice de confiance (inversement proportionnel à la tolérance aux faux positifs α), on s'attendrait plutôt à voir une courbe en cloche avec un maximum vers la gauche.

Figure 9: GENIE3 Random Forest

Figure 10: GENIE3 Extra Trees

Figure 11: Fold enrichment par nombre d'arcs

Pour GENIE3 nous itérons sur le nombre d'arcs pris en compte, des arcs avec les meilleurs indices de confiance à ceux ayant les plus mauvaises. Nous trouvons les optimums 39 arcs pour Random Forest (score: 3.088989) et 9 pour Extra Tree (score: 3.346405).
Rappelons que notre réseau de référence comporte 68 arcs. C'est intéressant de noter que nos réseaux "optimums" ont tout deux un nombre d'arc (beaucoup) plus faible que le réseau de référence.
Concrètement les arcs de la méthode Random Forest ont de meilleures indices de confiances [0.52 en moyenne] que la méthode Extra Trees [0.35 en moyenne] cependant le second a un meilleur score de fold enrichment.
Rappelons aussi que le réseau de référence est incomplet, les arcs prédits ne sont pas forcément faux. Cette fonction de scoring pondère trop les faux négatif, nos réseaux ont donc tendance à être très petit car les arcs les plus probables ne sont pas tous dans le réseau de référence car notre base de donnée n'a aucune informations sur ces gènes, tentons de corriger cela.

Depuis les données expérimentales d'interaction que nous possédons, chaque TF est en moyenne en interaction avec environ 3.58 autres TF. En faisant l'hypothèse que les marks d'histones se comportent comme les TF que nous observons dans nos données et donc ont le même nombre d'interaction avec d'autres TF. On estime alors qu'un réseau de 32 noeuds devrait avoir 114.53 arcs. Ce qui semble être raisonnable puisque cela représente seulement 11% de la densité maximale (32*32 = 1024), sachant que les réseaux typiques sont relativement peu dense. Nous introduisons une pénalité de socre sur les réseaux dont le nombre d'arc s'écarte de cette estimation.

3.1 Score, Fold Enrichment

$$Score_{fe} = \frac{\#correct\_edges}{\#randomly\_correct\_edges} \quad (1)$$

$$Score_{fe} = \frac{\frac{TP}{(\#network\_edges).(\#dataset\_edges)}}{N} \quad (2)$$

$$Score_{fe} = \frac{TP.N}{(TP+FP).(TP+FN)} \quad (3)$$

TP: True positive, liaison predites par le réseau et les données expérimentales
FP: False positive, liaison prédite seulement par le réseau
FN: False négative, liaison absente de les bases de donnée
N: Toutes les liaisons possibles

3.2 Score, Fold Enrichment avec pénalité

$$Score_{penalty} = \frac{n_{arc\_est}}{1+(n_{arc\_pred}-n_{arc\_est})^2}.Score_{fe} \quad (4)$$

Ce n'est que le fold enrichment multiplié à un facteur de pénalité calculé sur l'écart entre le nombre d'arcs prédits et nombre d'arc estimé. On multiplie avec le nombre d'arc estimé pour ne pas avoir de valeur trop petite.

4 Results

4.1 Based on Fold enrichment

Donner, nombre d'arcs, nombre d'autorégulation et scores

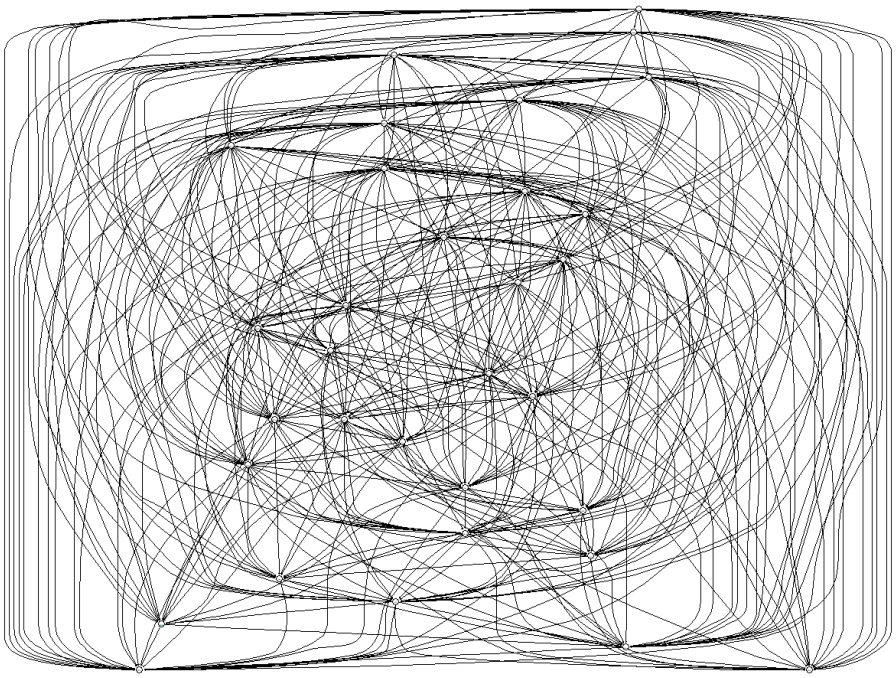

Figure 12: HC (score-based), 0 autorégulateur, 389 arcs, fold enrichment: 0.8517, penalty score: 0.001

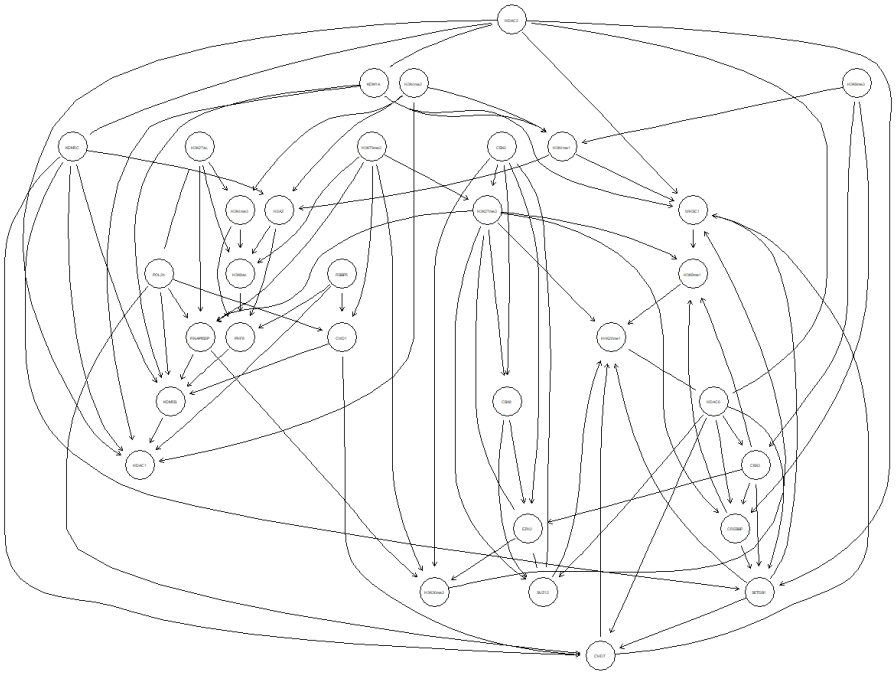

Figure 13: PC (constraint-based) α: 0.4, 0 autorégulateur, 96 arcs, fold enrichment: 2.039, penalty score: 0.687

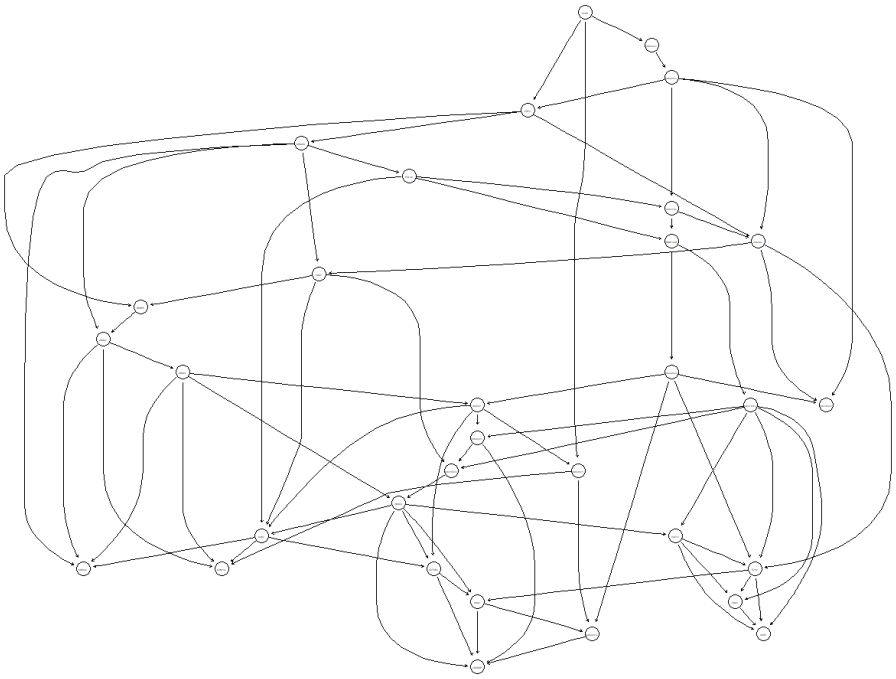

Figure 14: MMHC (hybrid) α: 0.05, 0 autorégulateur, 75 arcs, fold enrichment: 1.3514, penalty score = 0.116

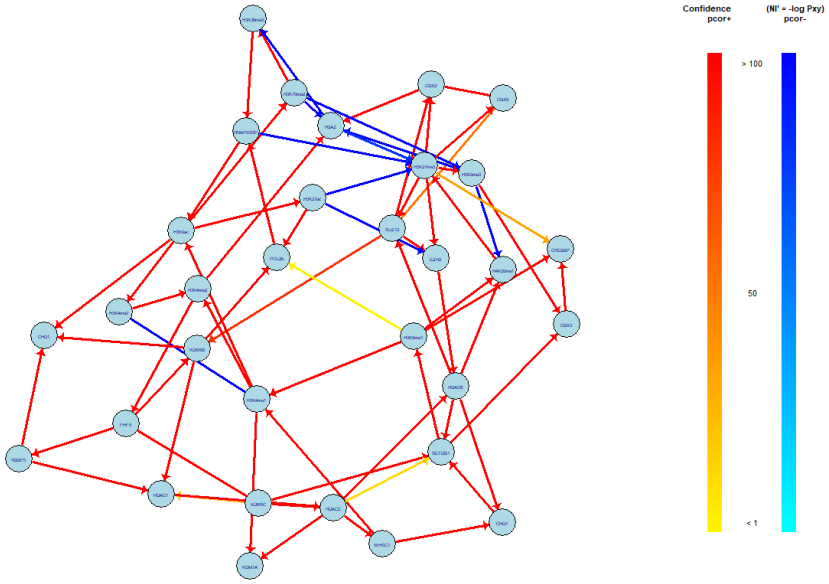

Figure 15: miic (constraint-based), 0 autorégulateur, 140 arcs, foldenrichment: 1.721, penalty score: 0.304

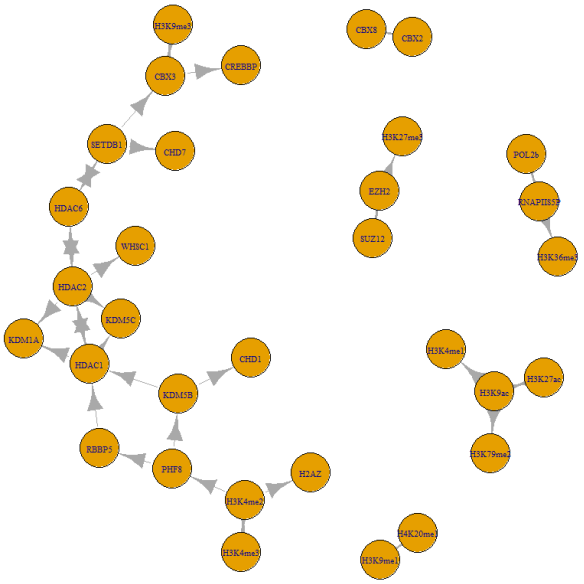

Figure 16: GENIE3 Random Forest (feature importance), 0 autorégulateur, 39 arcs, foldenrichment: 3.088989, penalty score: 0.062

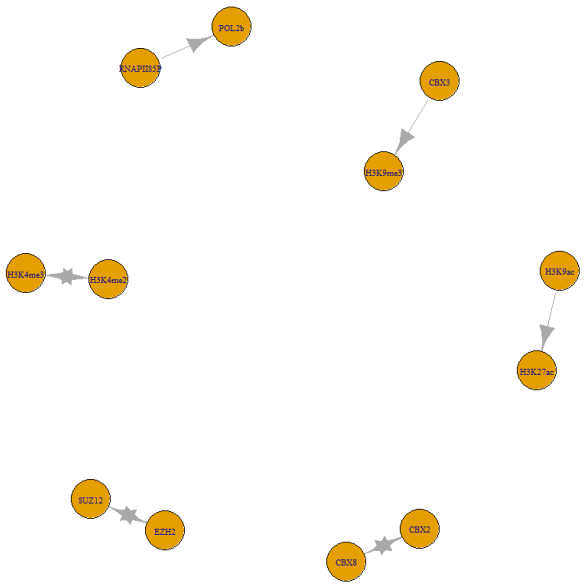

Figure 17: GENIE3 Extra Trees (feature importance), 0 autorégulateur, 9 arcs, foldenrichment: 3.346405, penalty score: 0.034

Première remarque, on ne trouve aucune autorégulation (ce qui est tout à fait normal, les algorithmes sont conçus pour éviter cela), pourtant cela représente 17% des interactions dans nos données expérimentales.

GENIE3 a les meilleures performances basées sur le fold enrichment, suivi des algorithmes constraint-based. Pourtant les réseaux GENIE3 semblent très éloignés de la réalité.

4.2 Based on Penalty score

On ne parlera que de PC et GENIE3, les autres algorithmes n'ont pas vu de changements significatifs sur leurs scores.

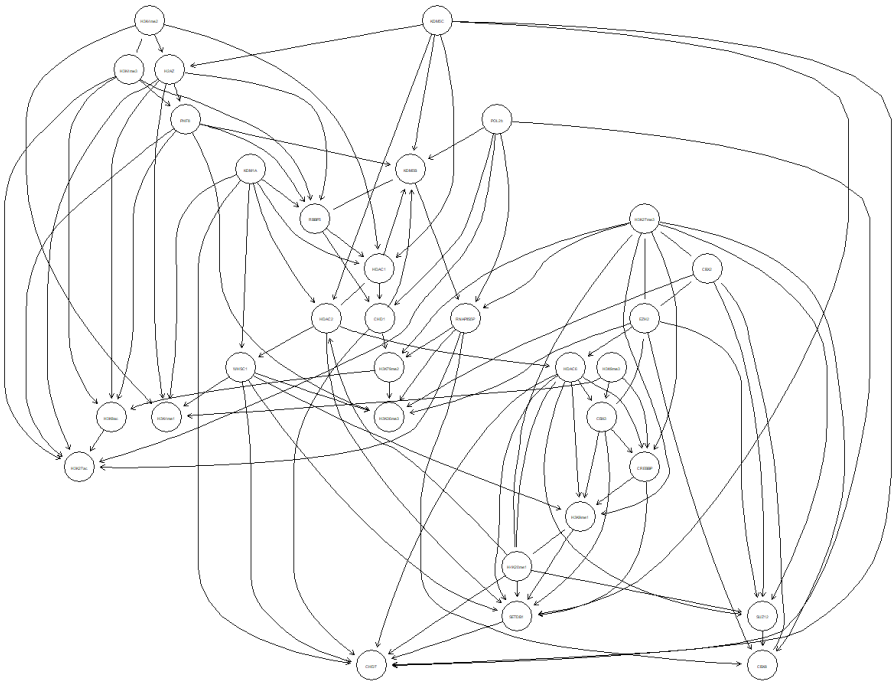

Figure 18: PC (constraint-based) α: 0.8, 0 autorégulateur, 112 arcs, fold enrichment: 1.882, penalty score: 29.13

Pour PC, nous remarquons que le graph est assez similaire (3 hubs) et même forme générale. Le score fold enrichment est aussi resté plutôt élevé, mais le penalty score est bien plus élevé.

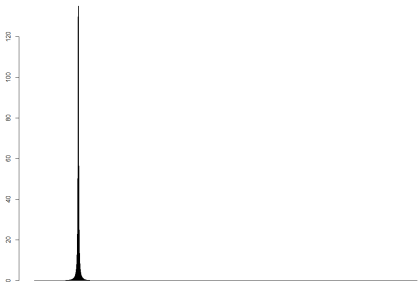

Figure 19: GENIE3 Random Forest

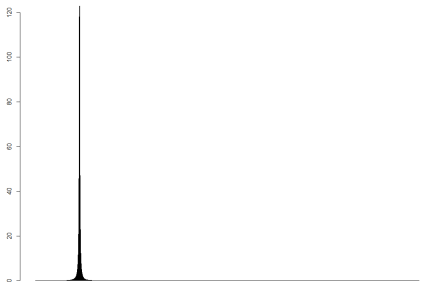

Figure 20: GENIE3 Extra Trees

Figure 21: Fold enrichment avec pénalité par nombre d'arcs

Ici on observe un biais énorme sur le nombre d'arc. N'oublions pas que score de

pénalité est aussi une heuristique pour tenter de contrer le manque d'informations expérimentales qui biaisais le score de fold enrichment, mais il ne faut pas non plus se fier entièrement au score de pénalité.

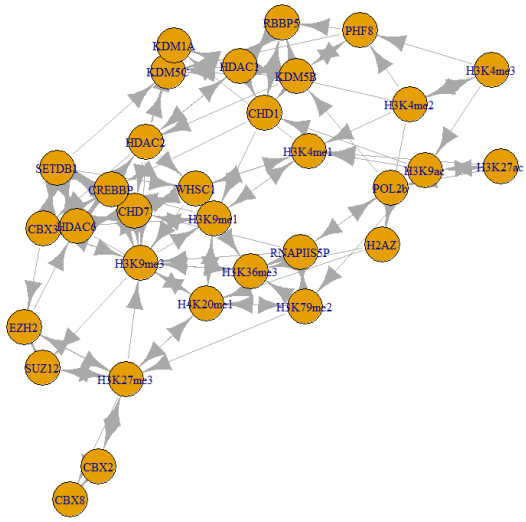

Figure 22: GENIE3 Random Forest (feature importance), 0 autorégulateur, 115 arcs, foldenrichment: 1.440409, penalty score: 135.1217

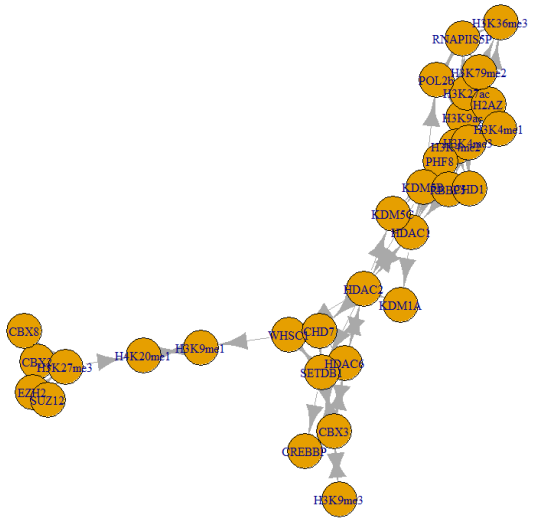

Figure 23: GENIE3 Extra Trees, (feature importance), 0 autorégulateur, 115 arcs, foldenrichment: 1.30946292, penalty score: 122.8379

GENIE3 (Random Forest) surclasse GENIE3 (Extra Trees) sur les 2 scores, rendant le second obsolète. De ce fait, il ne nous reste que 2 réseaux optimals, l'un avec un meilleur score de fold enrichment et l'autre avec un meilleur score de penalité.
Logiquement on se demande, qui croire? Le score de pénalité ou le score de fold enrichment? On a pu voir précédemment que le fold enrichment n'est pas totalement fiable mais le score de pénalité ajoute un biais important.

Subjectivement, on peut choisir en fonction de l'algorithmes sous-jacent, dans ce cas GENIE3 devrait théoriquement avoir les meilleurs résultats comme l'algorithme a été conçu spécifiquement pour générer des réseaux de régulation.
Objectivement, on peut optimiser en fonction des score de pénalité (comme c'est le cas ici) puis choisir celui avec le meilleur fold enrichment, ce qui nous indique le réseau PC.

4.3 Retirer les arcs d'autorégulation

Avons-nous un intérêt à garder les arcs d'autorégulation du réseau de référence? Ces arcs sont inutiles puisque les algorithmes évitent activement ce cas de figure, car probablement éloigné de la réalité (interaction indirect, un TF n'intéragit

pas avec soi même).

En retirant les arcs d'autorégulation nous arrivons à 56 interactions avec les mêmes noeuds. Ce qui fait 2.9473684210526314 interactions en moyenne, donc le réseau devrait avoir 94.3157894736842 arcs en moyenne. Toujours pas de changement significatif pour HC et miic, mmhc.

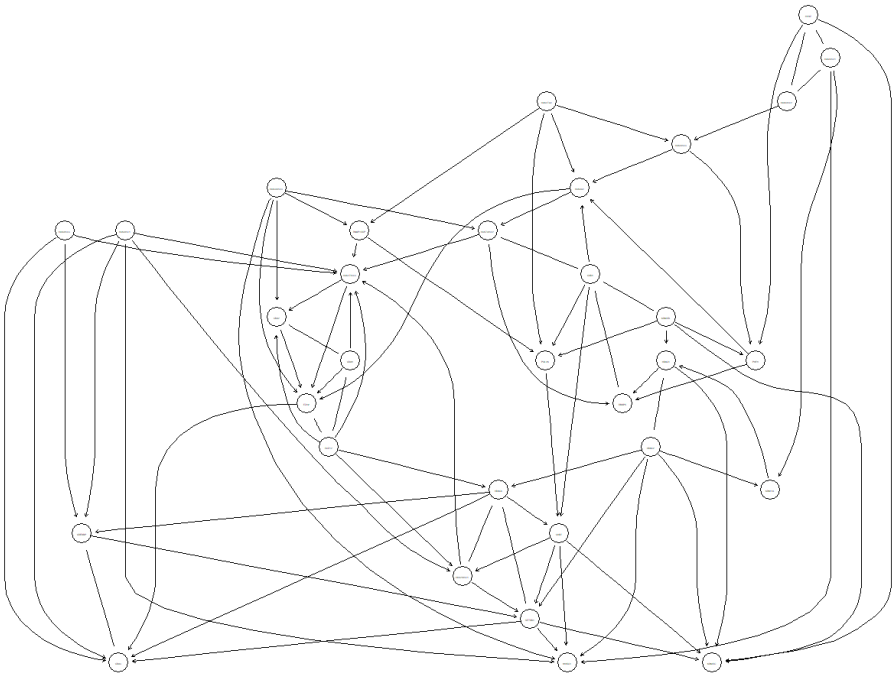

Figure 24: PC (constraint-based) α: 0.1, 97 arcs, fold enrichment: 1.902, penalty score: 122.2

Pour PC, les paramètres sont plus cohérent car taux de faux positif α à 0.8 est aberrant. Nous retrouvons un nombre d'arc similaire à lorsque nous classions sur le fold enrichment seul.

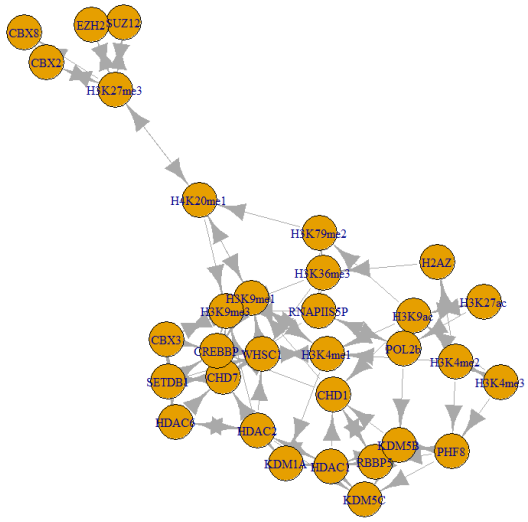

Figure 25: GENIE3 Random Forest, 97 arcs, fold enrichment: 1.602003, penalty score: 137.392900

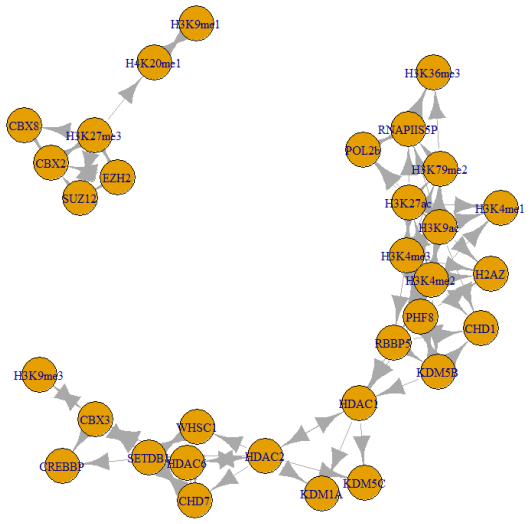

Figure 26: GENIE3 Extra Trees, 97 arcs, fold enrichment: 1.44180225, penalty score: 123.65360983

Pour GENIE3, les scores sont un peu meilleur. La méthode Extra Trees prédit un réseau à 2 composantes tandis que tous les autres algorithmes s'accordent à dire qu'il s'agirait d'un réseau à une grande composante.

5 Conclusion

D'après nos métriques, les algorithmes PC et GENIE3(Random Forest) prédisent les réseaux les plus fidèles. Cependant nous savons qu'il manque des entrées dans notre base de donnée (il manque les données histone) pour permettre de reconstruire un réseau de référence proche de la réalité. De ce fait cela nuit aux résultats de nos fonctions de scoring dépendantes des informations expérimentales. Afin d'améliorer nos résultats, il faut remplir les bases de données expérimentales.

References