



UNIVERSITÉ  
**PARIS**  
**DESCARTES**

UNIVERSITÉ PARIS DESCARTES

MLSD 2 2018-2019

---

## Méthodologie de la recherche

---

*Professeur :*

Séverine Affeldt  
LIPADE

*Auteurs :*

Joseph Gesnouin  
Yannis Tannier

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Mécanismes biologiques et développement de tumeurs</b>	<b>3</b>
<b>3</b>	<b>Materiel et méthodes</b>	<b>4</b>
3.1	Algorithmes de reconstruction de graphes . . . . .	4
3.1.1	Hill-Climbing approach . . . . .	5
3.1.2	Algorithme PC . . . . .	5
3.1.3	MIIC / 3off2 . . . . .	6
3.2	Visualisation: Fruchtermanreingold . . . . .	7
<b>4</b>	<b>Exploration des données</b>	<b>8</b>
<b>5</b>	<b>Création des graphes</b>	<b>10</b>
5.1	Hill-climbing . . . . .	11
5.2	Algorithme PC . . . . .	15
5.2.1	alpha 0.01 . . . . .	16
5.2.2	alpha 0.3 . . . . .	19
5.3	MIIC / 3off2 . . . . .	20
5.3.1	MIIC Shuffle=0 Threshold=0 . . . . .	21
5.3.2	MIIC Shuffle=1 Threshold=0.001 . . . . .	21
5.3.3	MIIC Shuffle=10000 Threshold=0.001 . . . . .	22
5.3.4	MIIC Shuffle=1 Threshold=0.00001 . . . . .	22
<b>6</b>	<b>Conclusion</b>	<b>26</b>
	<b>Bibliographie</b>	<b>27</b>

## 1 Introduction

Le but de ce projet est de reconstruire un réseau de gênes et de l'analyser afin de mieux comprendre l'importance de l'alteration de certains gênes dans le développement des cancers du sein ainsi que leur influence sur la ploïdie des cellules tumorales.

Nous avons souhaité, dans un premier lieu, nous documenter sur les mécanismes d'apparition du cancer: comment certains cancers se développaient au niveau cellulaire afin de mieux comprendre le jeu de données sur lequel nous travaillions.

Dans un second temps, nous nous sommes intéressés aux différentes méthodes de construction de graphes existantes dans la littérature, ainsi que leurs différences. Certaines d'entre elles étant basées sur l'optimisation de contraintes, d'autres sur un critère de score, et certaines plus hybrides, combinant alors des caractéristiques des deux précédentes.

Puis, après avoir sélectionné une méthode représentative pour chacun des types de méthodes présentées, nous avons regardé de plus près le jeu de données qui nous avait été attribué: sparsité, discréétisation de variables, outliers, valeurs manquantes et avons finalement réalisé une étude comparative des résultats obtenus pour les méthodes suivantes: Hill-Climbing, PC, et Miic sur le jeu de données cosmicCancer.

## 2 Mécanismes biologiques et développement de tumeurs

Le cancer se caractérise par une prolifération anormale de cellules dans un tissu. Un être humain est doté d'environ 100 000 milliards de cellules, à tout moment de notre vie, ces cellules meurent et d'autres sont créées pour les remplacer. Ces nouvelles cellules apparaissent grâce à la mitose, division d'une cellule mère en deux cellules filles strictement identiques génétiquement.

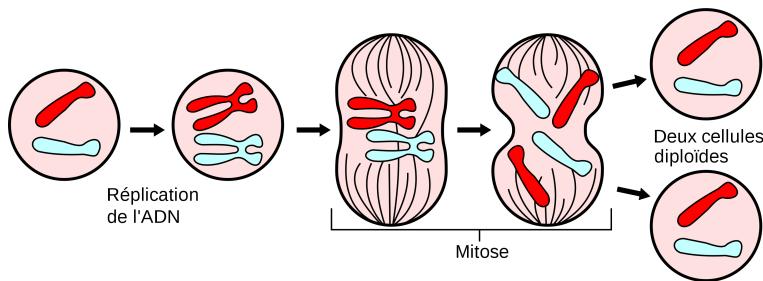


Figure 1: La mitose permet la formation de deux cellules filles strictement identiques génétiquement à la cellule mère.

Chez un être humain en bonne santé, un équilibre se crée entre le nombre de cellules qui meurent et celles qui se créent par mitose. La division cellulaire est un processus très encadré, très régulé: il existe des dizaines de mécanismes qui sont présents au sein du vivant afin de la contrôler et la surveiller pour maintenir cet équilibre entre le nombre de cellules qui meurent et celles qui apparaissent. Si par malheur, un jour, tous ces mécanismes dans une cellule sont innactifs, celle-ci peut échapper à la régulation, se mettre à se diviser de façon incontrôlée et créer une tumeur.

Derrière chacun de ces mécanismes de contrôle il y'a des gènes. Si un des mécanismes de contrôle se met à ne plus fonctionner de manière adéquate, c'est qu'un des gènes qui le pilote est altéré. En un certain sens, le cancer est une maladie génétique présente dans certaines cellules.

Pour qu'une cellule devienne potentiellement cancéruse, il faut que plusieurs mutations s'accumulent au fur et à mesure du temps, chacune des mutations alterant alors le bon fonctionnement de chacun des mécanismes de contrôles. Un cancer se caractérise alors par la séquence de mutation qui l'ont engendré. On retrouve cependant des régularités, des séquences de mutations qui semblent en favoriser d'autres lors d'un cancer. Par exemple, le gène tp53 impliqué dans plusieurs des mécanismes de contrôle, subirait une mutation dans près de la moitié des cancers.

C'est d'ailleurs ce qui lui a valu comme surnom "le gardien du génome". Directement lié au développement du cancer, l'humain ne possède que très peu de copies de ce facteur de transcription, c'est d'ailleurs ce qui fait que nous sommes relativement sensibles au développement de cancers. À contrario, les éléphants disposent de 15 à 20 copies de ce facteur de transcription dans leur génome, ce qui en fait des animaux peu sujets au développement de cancer. Ils sont d'ailleurs souvent sujets d'études à ce propos: *Sulak et al.*[2].

La ploïdie est le nombre d'exemplaires dans une cellule de jeux des chromosomes du génome. Dans notre cas d'utilisation sur le dataset *cosmicCancer* nous nous intéresserons à deux types de ploïdie:

- Une cellule est **diploïde** si elle possède 2 jeux, donc  $2 n$  chromosomes, organisés en  $n$  paires. Une cellule tumorale disposant du même nombre de chromosomes que la normale aura tendance à croître lentement et être moins agressive que les autres.
- Une cellule est **triploïde**: si elle possède 3 jeux , donc  $3 n$  chromosomes, organisés en  $n$  paires. Lorsqu'une cellule cancéreuse se divise trop rapidement, des erreurs lors de la distribution des chromosomes peuvent arriver, il en resultera des cellules avec trop de chromosomes. Ces types de cancer peuvent être plus agressifs que les autres.

Afin de mieux comprendre les mécanismes de développement du cancer et de l'importance de l'alteration de certains gènes dans la ploïdie de certains cancers, nous avons donc souhaité recréer un réseau de gènes grâce à plusieurs méthodes de création de graphes.

### 3 Materiel et méthodes

#### 3.1 Algorithmes de reconstruction de graphes

Ces dernières années, les réseaux bayésiens ont été utilisés dans de multiples domaines, l'un d'entre eux étant l'analyse des expressions de gènes.

La grande dimensionalité des jeux de données manipulés aura forcé les chercheurs à développer des algorithmes se concentrant sur la réduction de la complexité computationnelle de ces algorithmes tout en apprenant le bon réseau.

Les réseaux bayésiens sont des graphes dont les noeuds représentent des variables aléatoires et les arrêtes représentent une probabilité de dépendance entre ces deux noeuds.

De ces algorithmes, nous pouvons distinguer en trois approches:

- **constraint based:** La majorité de ces algorithmes utilisent des tests statistiques d'indépendance conditionnelle afin de trouver la structure du réseau bayésien.
- **score-based:** Ces méthodes se basent sur l'optimisation d'heuristiques afin de noter la structure des réseaux proposés grâce à un score de fitting.
- **hybrides:** Ces méthodes combinent les propriétés des deux précédentes: elles utilisent l'indépendance conditionnelle pour réduire l'espace de recherche et combinent celle-ci avec le score du réseau afin de trouver le meilleur réseau dans l'espace réduit.

Ce projet nous aura permis de manipuler plusieurs méthodes de construction de graphe aux approches différentes:

### 3.1.1 Hill-Climbing approach

L'algorithme de Hill-Climbing, est une des méthodes basée sur le score. Celle-ci prend en entrée trois paramètres: une configuration, une fonction qui pour chaque configuration donne un ensemble de configurations voisines, et une fonction de coût permettant d'évaluer chaque configuration. La méthode consiste simplement à partir de la configuration initiale, d'évaluer les solutions voisines, choisir la meilleure puis recommencer l'opération jusqu'à arriver à un optimum local. En d'autres termes, la méthode de hill-climbing n'est autre qu'une recherche gloutonne dans l'espace des graphes dirigés du meilleur graphe minimisant la fonction de coût. Plusieurs optimisations sont possibles comme l'approche tabu, permettant alors d'éviter d'être coincé dans des optimums locaux.

Il existe une variante de HC hybride: la méthode Max-Min Hill-Climbing, combinant l'algorithme MMPC pour réduire l'espace de recherche et HC pour trouver le réseau optimal dans l'espace réduit. Néanmoins, nous avons préféré nous intéresser à d'autres méthodes hybrides telles que MIIC au lieu d'implémenter cette méthode hybride.

### 3.1.2 Algorithme PC

L'algorithme PC, est une des premières méthodes basée sur l'optimisation de contraintes. Celui-ci a été développé afin de représenter des graphes orientés acycliques en se basant sur l'inference causale.

Un des paramètres que nous avons pu manipuler lors de l'implémentation de cet algorithme est le degré de signification alpha, représentant l'erreur de type I qui correspond au rejet de l'hypothèse nulle pour le test statistique d'inférence (faux-positif).

Il existe souvent plusieurs DAGs distincts représentant exactement le même ensemble de relations d'indépendance. Cela est du au fait qu'il est impossible de déterminer de quel type de V-structure nous disposons.

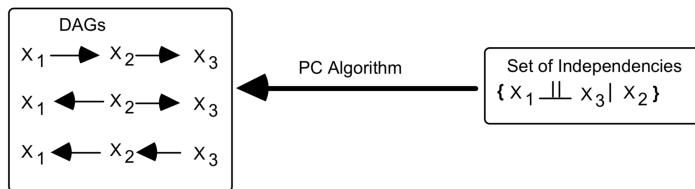


Figure 2: Différents DAGs provenant de la même relation d'indépendance

Ce phénomène explique en partie pourquoi nos résultats et leurs directions pour cet algorithme seront parfois à prendre avec des pincettes: PC arrivant à capturer la notion de V structure mais n'arrivant pas à diriger les arrêtes comme souhaité ou attendu.

La complexité de cet algorithme est polynomiale voire exponentielle pour des mauvais scénarios. Le temps d'execuition de cette méthode étant directement corrélé à la taille du dataset, il existe d'autres méthodes supposées plus rapides comme FCI ou encore RFCI, que nous avons brièvement utilisé afin de manipuler plus rapidement le jeu de données sur des algorithmes à optimisation de contraintes.

### 3.1.3 MIIC / 3off2

Nous avons souhaité essayer certaines méthodes définies comme hybrides combinant alors les approches constraint based et score based en utilisant directement les méthodes de création de graphes du package miic: Multivariate Information based Inductive Causation *Verny and al.*[1]

Miic apprend en découvrant progressivement la contribution indirecte des variables. De cette manière Miic capture les effets directs et indirects entre les variables potentiellement liées. En partant d'un graphe complet, la méthode enlève de manière itérative les arrêtes considérées comme superflues. Celles-ci sont considérées comme superflues dès qu'une contribution provenant d'un chemin indirect plus importante

est découverte. Les arrêtes restantes étant alors orientées en se basant sur la causalité calculée.

En se basant sur la décomposition de l'information mutuelle qui est toujours réalisable, Miic commence par collecter les contributions les plus probables entre trois noeuds, puis les enlève de l'information initiale des informations entre deux noeuds jusqu'à décider d'une possible indépendance structurelle entre les deux noeuds.

Cependant, à partir du moment où l'algorithme coupe une arrête qu'il considère comme superflue alors qu'elle ne l'est pas, l'itérativité de l'algorithme fait que la représentation finale en sera affectée.

Miic se déroule en trois étapes distinctes:

- Trouver un graphe non orienté en prenant en compte les variables latentes de l'information mutuelle.
- Enlever les arrêtes faibles selon un critère de confiance.
- Orienter les arrêtes restantes en fonction de la signature de la causalité.

Lors de ce rapport, nous avons pu essayer de jouer avec les hyperparamètres de la fonction MIIC proposée par *Verny and al.*[1]:

- **confidenceShuffle**: utilisé pour spécifier le nombre de mélange du jeu de données à réaliser afin d'évaluer le niveau de confiance de chacune des arrêtes inférées.
- **confidenceThreshold**: utilisé pour filtrer les arrêtes les moins probables après la construction du graphe non orienté. En modifiant ce paramètre, le nombre d'arrêtes conservées pour le graphe final variera en fonction du niveau de laxisme que l'on choisit.

### 3.2 Visualisation: Fruchtermanreingold

Notre jeu de données étant assez conséquent, nous avons eu recourt à un algorithme de visualisation de graphe permettant une meilleure visibilité de la sémantique de celui-ci grâce à un système de force appliqués entre les noeuds et les arrêtes.

Le but de l'algorithme Fruchtermanreingold est de positionner les noeuds d'un graphe dans un espace de dimension deux, de telle sorte que toutes les arrêtes soient plus moins de taille égale en se croisant le moins possible.

Une fois l'attribution des forces à chacun des composants du graphe réalisée (noeuds et arrêtes), le comportement du graphe peut être simulé comme un système physique.

Les forces sont appliquées aux noeuds, les rapprochant ou les éloignant les un des autres. En répétant itérativement ce procédé, le graphique arrive à un état d'équilibre physique, la position de chacun des composants du graphe est alors stable. Une fois l'équilibre atteint, on affiche le graphe tel quel.

## 4 Exploration des données

Le jeu de données *cosmicCancer* contient 807 exemples incluant les niveaux d'expression de 91 gènes, sélectionnés grâce à plusieurs études préalables sur l'impact de ces gènes dans le développement des cancers du sein. L'information sur le statut d'un gène pour sa mutation est binaire et son niveau d'expression est categoriel: sous-exprimé, normal ou sur-exprimé.

Name	Value
Rows	807
Columns	176
Discrete columns	175
Continuous columns	1
All missing columns	0
Missing observations	8
Complete Rows	799
Total observations	142,032
Memory allocation	675.7 Kb

Figure 3: Information basique du jeu de données cosmicCancer

En plus de ces informations sur la mutation et le niveau d'expression de chaque gène, nous disposons de l'information de la ploïdie de chacun des exemples. Au sein des 807 exemples, 401 des cellules cancéreuses sont diploïdes et 398 triploïdes. Les 9 exemplaires restants ne disposant d'aucune valeur à propos de leur ploïdie, nous ne les avons pas considérés comme des exemplaires complets et donc ne les avons pas traités.

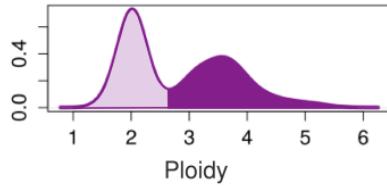


Figure 4: Distribution de la ploïdie non discrétisée: cosmicCancer

Initialement, la ploïdie était une valeur distribuée selon deux gaussiennes: dans un soucis de simplification, celle-ci aura été discrétisée selon un critère simple: toutes les cellules dont la ploïdie était inférieure à 2.7 auront été classées comme diploïdes et les autres triploïdes.

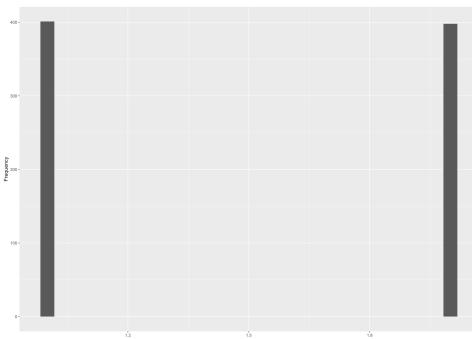


Figure 5: Distribution de la ploïdie discrétisée: cosmicCancer

Nous disposions donc dans le jeu de données, de plusieurs informations plus ou moins similaires pour chacune des cellules cancéreuses: les gênes étaient présentés en minuscule pour prévenir qu'ils avaient muté, l'expression de ces mêmes gênes était en majuscule et enfin leur ploïdie discréditée si elle avait été calculée. Dans un souci de clareté au niveau des graphes, nous avons souhaité projeter ces gênes mutés en rose, leur expression en rouge et la ploïdie en bleu.

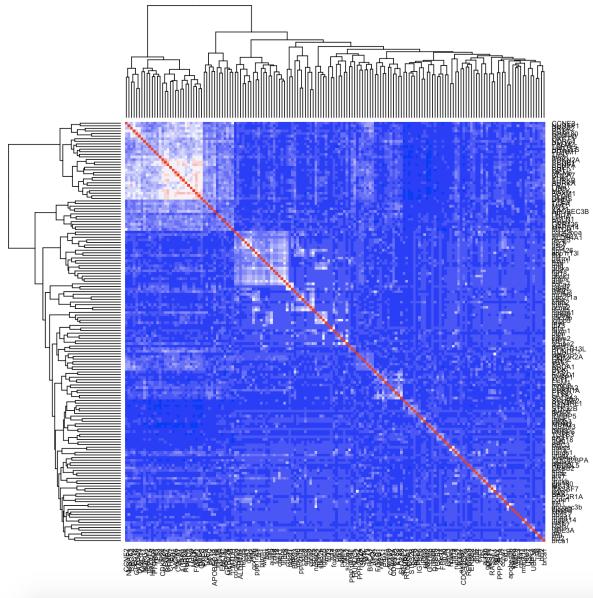


Figure 6: Matrice de corrélation du jeu cosmicCancer

Une analyse plus exhaustive de la fréquence de chacune des variables ainsi que d'une ACP sur le jeu de données est disponible [ici](#).

Grâce à celle-ci nous pouvons remarquer que la matrice est extrêmement vide, celle-ci étant sparse à 95%. Cette information nous permet donc d'être plus sereins vis-à-vis de l'utilisation d'algorithmes assez lourds comme PC qui d'après *Kalisch*[3] semble bien se dimensionner avec des matrices sparses de grande dimensions malgré une complexité polynomiale voir exponentielle.

L'ACP réalisée en annexe nous permet de noter que, bien que relativement faible au niveau de la variance totale, l'axe 1 semble utiliser pour sa relation linéaire la majorité des variables en minuscules et l'axe 2 semble utiliser au même titre les variables en majuscules.

## 5 Crédit des graphes

Une fois cette exploration préalable des données réalisée, nous nous sommes donc atelés à la construction des graphes selon les trois familles de méthodes présentées.

## 5.1 Hill-climbing

Afin de réorganiser la matrice sous forme de graphe afin de capturer les altérations de gènes susceptibles d'aider à la prolifération de cellules cancéreuses, nous avons du traiter notre matrice d'entrée:

```
df2=hc(cosmicCancer) ###Marche pas, cherche pk les NA g??nent
df1=cosmicCancer[complete.cases(cosmicCancer),]
head(df1);is.na(df1)
df1$Ploidy <- sapply(df1$Ploidy, as.factor) ### On change le type de Ploidy qui pouvait faire bugger
dim(df1)
setdiff(rownames(cosmicCancer), rownames(df1)) ###on regarde quelles cellules ??taient vides
df1=df1[,apply(data.matrix(df1), 2, var, na.rm=TRUE) != 0] ###On vire les colonnes constantes
df2=hc(df1) #on applique enfin notre hillclimbing
```

Figure 7: Pré-traitement du jeu cosmicCancer

Ainsi, nous avons du retirer du jeu de données les cellules dont certaines valeurs étaient manquantes: typiquement les 8 cellules dont la ploïdie n'était pas spécifiée: 90, 91, 252, 333, 419, 540, 700, 742. Ces cellules représentent environ 1% du jeu de données. Cela ne représente donc pas une partie conséquente de celui-ci et de ce fait, ce premier traitement ne biaisera pas l'intégrité de l'analyse en vue de sa faible importance sur le jeu de données total: d'une matrice  $807 \times 176$  nous passons à une matrice  $799 \times 176$ .

	bbc3	eglm1	tgbfb3	esml	igfbp5	fgf18	scube2	wisp1	flt1	hrasls	stk32b	rassf7	dck	melk	ext1	gnaz	ebf4	mtdh	pitrm1	qscn6l1	ccne2		
91	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n		
	ect2	cenpa	lin9	kntc2	mcm6	nusap1	orc6l	tspyl5	rundc1	prc1	rfc4	recql5	cda7	dtl	col4a2	gpr180	mmp9	gpr126	rtn4rl1	diaph3			
91	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n		
	cd42bp1	palm2	aldh4a1	aytl2	oxct1	peci	gmgs	gstm3	slc2a3	foxm1	erb2	esr1	tp53	rb1	myc	jun	cdkn2a	bcl2	tp73	lats2	mapk14		
91	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n		
	cdkn1a	chek1	aurkb	aurka	brca1	brca2	dusp5	mst1	ppp1r13l	birc3	tgfa	ets1	ets2	hif1a	ldha	foxo1	ndrg1	ppp2r1a	ppp2r2a	ccne1			
91	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n		
	apobec3b	pten	mdm2	usp7	ube3a	spdye7p	plk1	bax	met	bbc3	eglm1	tgbfb3	esml	igfbp5	fgf18	scube2	wisp1	flt1	hrasls				
91	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n		
	STK32B	RASSF7	DCK	MELK	EXT1	GNAZ	MTDH	PITRM1	CCNE2	ECT2	CENPA	LIN9	MCM6	NUSAP1	TSPYL5	RUNDCL1	PRC1	RFC4					
91	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal		
	RECQL5	CDCA7	DTL	COL4A2	GPR180	MMP9	GPR126	RTN4RL1	DIAPH3	CDC42BP1	PALM2	ALDH4A1	OXCT1	GMPS	GSTM3	SLC2A3	ERBB2						
91	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	over	normal	normal	normal	normal	normal	normal		
	ESR1	TP53	RB1	MYC	JUN	CDKN2A	BCL2	FOXM1	BRCA1	TP73	LATS2	MAPK14	CDKN1A	CHEK1	BRCA2	AURKB	AURKA						
91	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal		
	APOBEC3B	DUSP5	MST1	PPP1R13L	BIRC3	TGFA	ETS1	ETS2	HIF1A	LDHA	FOXO1	NDRG1	PPP2R1A	PPP2R2A	CCNE1	PTEN	MDM2						
91	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	normal	under	normal	normal	over		
	USP7	UBE3A	PLK1	BAX	MET	Ploidy																	
91	over	normal	normal	normal	normal	NA																	

Figure 8: Exemple d'une cellule du jeu de données cosmicCancer: ploïdie manquante

La méthode Hill-Climbing nécessitait également d'avoir des variables à variance non constante. Nous avons donc réalisé un second traitement sur cette matrice privée des 8 cellules pour enlever les variables qui disposaient toutes des mêmes valeurs:

nous avons donc retiré du jeu de données 14 gènes dont la variance était constante: *esm1*, *ebf4*, *qscn6l1*, *cenpa*, *kntc2*, *orc6l*, *aytl2*, *peci*, *gstm3*, *cdkn2a*, *cdkn1a*, *foxo1*, *ppp2r2a*, *spdye7p*. Ces 14 gènes ayant pour les 799 cellules restantes la même valeur, il n'était pas intéressant de les garder. Notre appel de la méthode de hill-climbing se faisant alors sur un jeu de données  $799 \times 162$ .

Figure 9: Exemple d'une variable du jeu de données cosmicCancer dont la variance est constante.

Après ces pré-traitements, nous avons enfin pu appliquer l'algorithme HC: le graphique généré étant assez lourd, nous avons fait le choix de représenter chaque type de variables par une couleur spécifique et avons omis de représenter les noeuds de degré nul qui ne pouvaient rien apporter au graphique à part de la compléxité. Sur les 162 noeuds restants, nous avons donc fait le choix de ne pas représenter 19 noeuds de degré nul.

```
> which(cigraph::degree(netZ)==0)
   bbc3    melk    ext1    mmp9 rtm4rl1  diaph3 aldh4a1  slc2a3    bcl2    tp73 mapk14    birc3    ets1    ets2 ccne1 ube3a
     1      13      14      32      34      35      38      41      49      50      52      61      63      64      69      74
BBC3 REQL5 MMP9
  78     106     111
```

Figure 10: Méthode hill-climbing: noeuds de degré nul

Grâce à la méthode de visualisation de fruchtermanreingold, nous obtenons le graphique suivant:

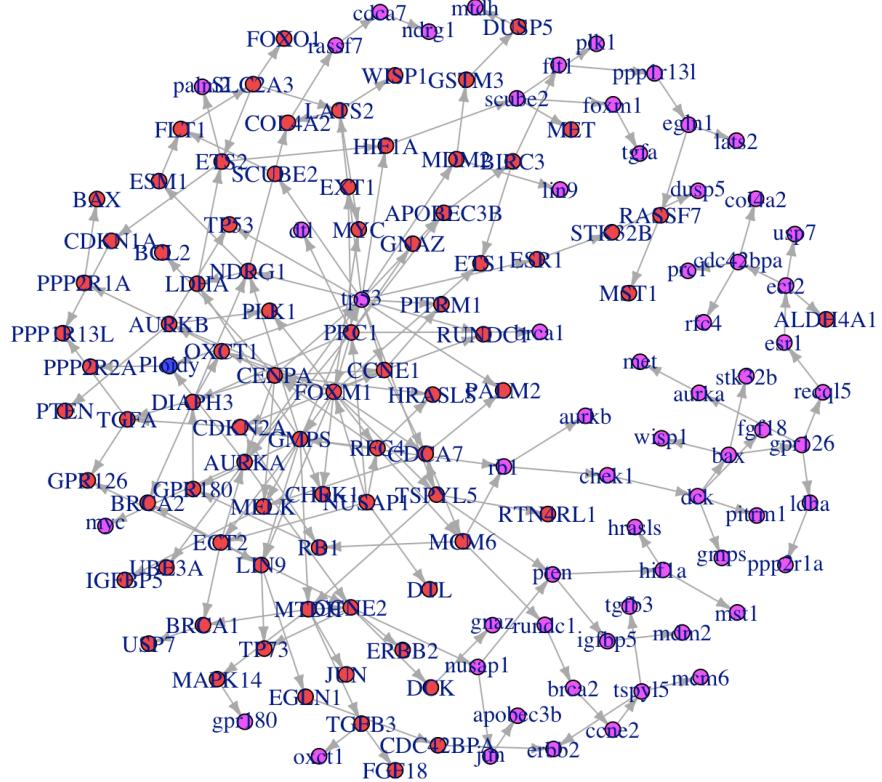


Figure 11: reconstruction de graphe cosmicCancer: méthode hill-climbing

Nous remarquons ici que la Ploïdie est directement monitorée par le gène *tp53* et sa mutation, celui-ci empêchant les mécanismes de contrôle de la mitose de fonctionner correctement. La méthode de Hill-Climbing nous apporte donc bien un lien direct de l'effet de la mutation de *tp53* dans les cancers de la poitrine et sur la triploidie des cellules cancéreuses, bien identifié par les derniers papiers traitant sur ce sujet.

En plus de cette relation bien connue du monde bio-informatique, nous avons pu identifier d'autres relations pour le moins classiques comme la sur-expression du gène AURKA et de son effet sur le cancer.

Plus généralement, Les variables directement liées à la ploïdie d'une cellule sont *tp53*, AURKA et PPP2R2A.

```
> E(net2) [ to("Ploidy") ]
+ 2/199 edges from 846f74e (vertex names):
[1] tp53 ->Ploidy AURKA->Ploidy
> E(net2) [ from("Ploidy") ]
+ 1/199 edge from 846f74e (vertex names):
[1] Ploidy->PPP2R2A
```

Figure 12: variables directement liée à la ploïdie: méthode hill-climbing

Après avoir identifié les gènes directement liés à la ploïdie d'une cellule tumorale, nous avons voulu identifier les gènes faisant figure de hub, et donc par définition étant éléments centraux d'expression et de mutations d'autres gènes:

CDKN1A	TGFA	nusap1	GPR180	MTDH	CDKN2A	RB1	MCM6	NDRG1
0.0002714542	0.0048268500	0.0076604274	0.0141198708	0.0235002343	0.0309108265	0.0370589591	0.0437141600	0.0458696433
BIRC3	ETS2	LATS2	ECT2	BRCA2	TSPYLS5	AURKB	NUSAP1	CDCA7
0.0583251293	0.0633655195	0.0649274852	0.0807301053	0.0921039780	0.1285531067	0.1338691384	0.1370548055	0.1462756209
AURKA	DIAPH3	RFC4	PRC1	CCNE1	CENPA	GMPS	FOXM1	tp53
0.1624071753	0.1722574354	0.1744653155	0.2510718261	0.2547987387	0.4750144378	0.6589621574	0.9787804456	1.0000000000

Figure 13: variables considérées comme hub du graphe: méthode hill-climbing

Nos résultats sont similaires à *Verny and al.*[1]. Ainsi, nous retrouvons donc les mêmes gènes moteurs du développement de cancer: AURKA, CENPA, GMPS, FOXM1, TP53...

Dernière information intéressante à regarder, nous avons souhaité identifier les noeuds et les arrêtes dont la centralité intermédiaire était la plus élevée. Celle-ci correspond au nombre de fois qu'un sommet/une arrête est sur le chemin le plus court entre deux autres noeuds du graphe. Un noeud/une arrête possède une grande intermédiarité s'ils ont une grande influence sur les transferts de données dans le réseau.

```
> get.edgelist(net2)[index,]
 [,1]      [,2]
 [1,] "dck"    "bax"
 [2,] "rb1"    "chek1"
 [3,] "chek1"  "dck"
 [4,] "bax"    "gpr126"
 [5,] "MCM6"   "rb1"
 [6,] "CDCA7"  "MCM6"
 [7,] "GPR180" "DIAPH3"
 [8,] "DIAPH3" "NDRG1"
 [9,] "RB1"    "GPR180"
 [10,] "AURKA"  "tp53"
```

Figure 14: Arrêtes considérées comme ayant la plus grande centralité intermédiaire: méthode hill-climbing

```
> V(net2)[index]
+ 10/162 vertices, named, from 846f74e:
[1] dck      gpr126  tp53    rb1     chek1   bax     MCM6    CDCA7   DIAPH3 NDRG1
```

Figure 15: Noeuds considérés comme ayant la plus grande centralité intermédiaire: méthode hill-climbing

## 5.2 Algorithme PC

De la même manière que pour la méthode de création de graphe précédente, nous avons du réorganiser notre matrice grâce à certains pre-traitements afin de pouvoir appliquer la méthode PC. Nous avons donc conservé la matrice pré-traitée précédemment et y avons apporté quelques modifications:

```
dt <- data.matrix(df1)-1
View(dt)
V <- colnames(dt)
summary(dt)

vect = c()
for(i in 1:162){
  vect = cbind(vect, length(unique(dt[,i])))
}
suffStat <- list(dm = dt, nlev = vect, adaptDF = FALSE)
```

Figure 16: Pré-traitement du jeu de données cosmicCancer: méthode PC

- Transformation de la matrice en un jeu de données numérique
- Changement des indices des variables afin de commencer à zero
- Calcul du nombre de valeurs possibles pour chaque variable
- Préparation de l'objet suffStat,

De la même manière que pour la méthode précédente et dans un souci de clareté, nous avons fait le choix de ne pas représenter les noeuds dont le degré était nul.

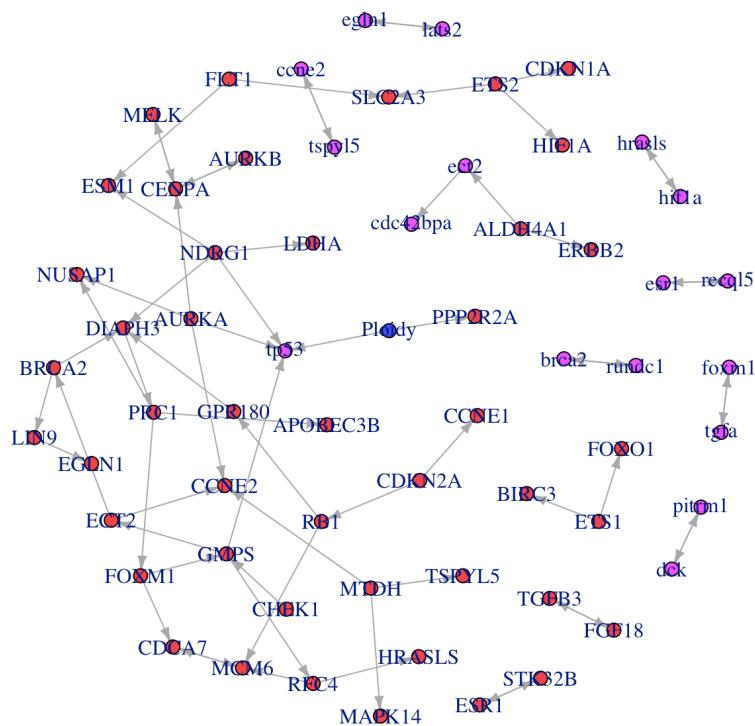
Après avoir fait varier plusieurs fois le paramètre alpha représentant l'erreur de type I acceptée et donc directement corrélé au nombre d'arrêts conservées et par conséquent du nombre de noeuds: plus alpha est élevé plus le nombre de noeuds aussi. Nous obtenons les résultats suivants:

### 5.2.1 alpha 0.01

Pour une valeur alpha de 0.01, 99 noeuds sont considérés comme ayant un degré nul:

> which(igraph::degree(net2)==0)																
bbc3	tgfb3	igfbp5	fgf18	scube2	wisp1	flt1	stk32b	rassf7	melk	ext1	gnaz	mtdh	lin9			
1	3	4	5	6	7	8	10	11	13	14	15	16	20			
mcm6	nusap1	prc1	rfc4	cdca7	dtl	col4a2	gpr180	mmp9	gpr126	rtn4rl1	diaph3	palm2	aldh4a1			
21	22	25	26	28	29	30	31	32	33	34	35	37	38			
oxct1	gmgs	slc2a3	erbb2	rb1	myc	jun	bcl2	tp73	mapk14	chek1	aurkb	aurka	brca1			
39	40	41	43	46	47	48	49	50	52	53	54	55	56			
dusp5	mst1	ppp1r13l	birc3	ets1	ets2	ldha	ndrg1	ppp2r1a	cgne1	apobec3b	pten	mdm2	usp7			
58	59	60	61	63	64	66	67	68	69	70	71	72	73			
ube3a	plk1	bax	met	BBC3	IGFBP5	SCUBE2	WISP1	RASSF7	DCK	EXT1	GNAZ	PITRM1	RUND1			
74	75	76	77	78	82	84	85	89	90	92	93	95	103			
RECQL5	DTL	COL4A2	MMP9	GPR126	RTN4RL1	CDC42BPA	PALM2	OXCT1	GSTM3	TP53	MYC	JUN	BCL2			
106	108	109	111	112	113	115	116	118	120	124	126	127	129			
BRCA1	TP73	LATS2	DUSP5	MST1	PPP1R13L	TGFA	PPP2R1A	PTEN	MDM2	USP7	UBE3A	PLK1	BAX			
131	132	133	141	142	143	145	152	155	156	157	158	159	160			
MET																
161																

Figure 17: Méthode PC alpha 0.01: noeuds de degré nul



Comme précédemment, la ploïdie est ici directement corrélée avec le gène tp53 et PPP2R2A mais dans le sens inverse: ceci est du au fait que l'algorithme PC aura bien identifié la V structure tp53, Ploidie, PPP2R2A mais orienté les arrêtes d'une manière différente.

```
> E(net2) [ to("Ploidy") ]
+ 0/69 edges from 943f2e1 (vertex names):
> E(net2) [ from("Ploidy") ]
+ 2/69 edges from 943f2e1 (vertex names):
[1] Ploidy->tp53    Ploidy->PPP2R2A
```

Figure 19: variables directement liée à la ploïdie: méthode PC 0.01

De la même manière avec une valeur de alpha aussi faible, certains des noeuds considérés comme des hubs par la méthode précédente semblent ne pas l'être ici: ainsi tp53 n'est ni considéré comme hub par notre graphe actuel, ni considéré comme noeud dont la centralité intermédiaire était la plus élevée du graphe:

ERBB2	MAPK14	CDKN1A	APOBEC3B	BIRC3	HIF1A	LDHA	FOXO1	PPP2R2A
0.000000e+00								
CCNE1	egl1n1	hrasls	dck	pitrm1	ccne2	ect2	tspyl5	rundc1
0.000000e+00	2.257082e-17							
recql5	foxm1	esr1	lats2	brca2	tgfa	hif1a	TGFB3	FGF18
2.257082e-17								
STK32B	LIN9	ESR1	DIAPH3	CHEK1	MCM6	RFC4	RB1	ALDH4A1
2.257082e-17	2.257082e-17	2.257082e-17	4.514163e-17	4.514163e-17	9.028327e-17	9.028327e-17	9.028327e-17	1.128541e-16
CDKN2A	ETS1	CDCA7	CENPA	FOXM1	ETS2	GPR180	FLT1	MELK
1.128541e-16	1.128541e-16	1.354249e-16	1.805665e-16	1.805665e-16	4.244432e-02	1.703867e-01	1.732257e-01	1.968021e-01
AURKB	BRCA2	PRC1	ECT2	MTDH	Ploidy	GMPS	NDRG1	AURKA
1.968021e-01	1.984051e-01	2.450232e-01	2.574371e-01	3.081012e-01	4.731071e-01	5.662156e-01	8.377584e-01	1.000000e+00

Figure 20: variables considérées comme hub du graphe: méthode PC 0.01

```
> get.edgelist(net2)[index,]
[,1]      [,2]
[1,] "ECT2"   "BRCA2"
[2,] "PRC1"   "FOXM1"
[3,] "GPR180" "DIAPH3"
[4,] "DIAPH3" "PRC1"
[5,] "GMPS"   "ECT2"
[6,] "GMPS"   "RFC4"
[7,] "RB1"    "GPR180"
[8,] "FOXM1"  "GMPS"
[9,] "BRCA2"  "LIN9"
[10,] "BRCA2" "DIAPH3"

> V(net2)[index]
+ 10/63 vertices, named, from 943f2e1:
[1] ECT2   LIN9   PRC1   RFC4   GPR180  DIAPH3  GMPS   RB1    FOXM1  BRCA2
```

Figure 21: Arrêtes et noeuds considérées comme ayant la plus grande centralité intermédiaire: méthode PC 0.01

Nous retrouvons néanmoins dans ces trois figures certains des gènes dont l'alteration et l'influence sur le cancer est avérée: GMPS, NDRG1, AURKA, FOXM1. Cependant, notre graphique avec cet alpha étant très fortement restreint, de par son nombre de noeuds affichés, nous avons souhaité réessayer la méthode PC en faisant varier le paramètre alpha afin d'obtenir un graphique plus significatif, disposant de plus de noeuds et potentiellement de relations que nous avions coupé dès le début qui auraient pu être porteuses d'information comme le gène tp53 dont l'algorithme PC avec la valeur alpha de 0.01 ne semble pas avoir mesuré l'importance. Afin de ne pas perdre notre temps à lancer des instances de l'algorithme PC pour des valeurs de alpha aléatoire, nous avons envisagé l'utilisation d'une variante de cet algorithme supposée plus rapide: RFCI afin de nous faire une idée plus précise du paramètre pour lequel PC pouvait être plus convainquant sans perdre trop de temps de calcul, dans les faits, les deux méthodes sont plutôt assez longues et nous n'avons noté que très peu d'optimisation du temps de calcul entre les deux.

En faisant varier l'alpha en l'augmentant sensiblement d'une puissance de 10, nous espérons voir apparaître plus de noeuds de degré non nul et donc un graphique moins extrême que celui présenté précédemment.

### 5.2.2 alpha 0.3

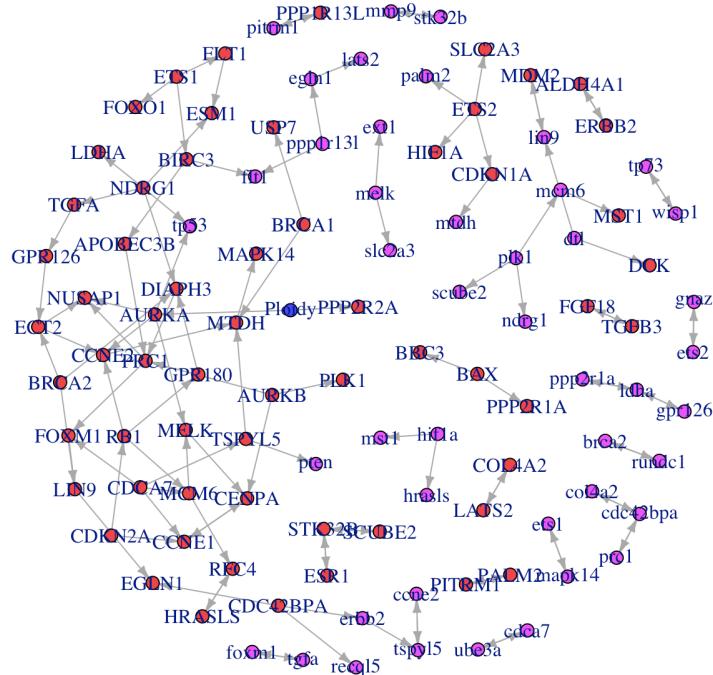


Figure 22: reconstruction de graphe cosmicCancer: méthode PC alpha 0.3

	USP7	PLK1	BAX	Ploidy	HRASLS	FOXM1	GPR126	CENPA	CDKN2A	FLT1	GPR180
0.00000000	0.00000000	0.00000000	0.00000000	0.03452441	0.04367742	0.04367742	0.06987926	0.08153198	0.13089330	0.17457071	
MCM6	BRCA2	CDCA7	PRC1	RB1	ECT2	NDRG1	AURKA				
0.20703661	0.26192554	0.33752127	0.38222344	0.47785536	0.57237995	0.78494336	1.00000000				

Figure 23: variables considérées comme hub du graphe: méthode PC 0.3

```
> E(net2) [ to("Ploidy") ]
+ 0/120 edges from 053ce3d (vertex names):
> E(net2) [ from("Ploidy") ]
+ 2/120 edges from 053ce3d (vertex names):
[1] Ploidy->AURKA   Ploidy->PPP2R2A
```

```
> E(net2) [ from("tp53") ]
+ 0/120 edges from 053ce3d (vertex names):
> E(net2) [ to("tp53") ]
+ 2/120 edges from 053ce3d (vertex names):
[1] AURKA->tp53 NDRG1->tp53
```

Figure 24: variables directement liée à la ploïdie et à tp53: méthode PC 0.3

```

[,1]      [,2]
[1,] "MTDH"   "MAPK14"
[2,] "CCNE2"   "MTDH"
[3,] "ECT2"    "CCNE2"
[4,] "LIN9"    "EGLN1"
[5,] "PRC1"    "FOXM1"
[6,] "GPR180"  "DIAPH3"
[7,] "GPR126"  "ECT2"
[8,] "DIAPH3"  "PRC1"
[9,] "RB1"     "GPR180"
[10,] "CDKN2A" "RB1"
[11,] "FOXM1"  "LIN9"
[12,] "APOBEC3B" "PRC1"

> V(net2)[index]
+ 10/107 vertices, named, from 053ce3d:
[1] MTDH CCNE2 ECT2 LIN9 MCM6 PRC1 GPR180 DIAPH3 RB1 FOXM1

```

Figure 25: Arrêtes et noeuds considérées comme ayant la plus grande centralité intermédiaire: méthode PC 0.3

Nous obtenons ici un graphe moins restreint, par exemple tp53 n'a pas été évincé et l'on peut noter que celui-ci dispose des mêmes liaisons que celles trouvées précédemment via l'algorithme de Hill-Climbing: par exemple la liaison entre la sur-expression d'AURKA et son influence sur tp53, ou encore l'influence de NDRG1, gène modulant la croissance cellulaire, sur tp53 en favorisant la progression des tumeurs et des métastases. Cependant, une majorité des arrêtes de tp53 n'ont pas survécu à notre indice de confiance et beaucoup des liaisons trouvées via l'algorithme de Hill-Climbing ont été coupées.

En ce qui concerne les hubs, les plus certains sont bien évidemment AURKA et NDRG1, nous aurions espéré trouver tp53, élément moteur de la déclaration de cancer mais il semble que nous ayons encore une fois été trop laxistes sur notre alpha.

Nous avons également calculé les noeuds et les arrêtes avec la plus grande centralité intermédiaire, comme pour la méthode Hill-Climbing, nous retrouvons des liaisons assez connues et assurées comme BRCA2, FOXM1...

### 5.3 MIIC / 3off2

Dans un premier temps, nous avons souhaité manipuler les hyperparamètres confidenceShuffle et confidenceThreshold afin de voir l'impact qu'ils pouvaient avoir sur le graphe généré et ensuite sélectionner le graphe qui nous convenait le mieux. Il s'avère que plus le threshold est faible, moins le nombre d'arrête sera élevé. Celui-ci correspondant à la confidence nécessaire pour filtrer les arrêtes les moins probables

après la première étape de l'algorithme. Plus le confidenceShuffle est élevé, plus le temps de calcul est élevé, mais celui-ci permet d'évaluer le niveau de confiance de chacune des arrêtes plus précisément.

### 5.3.1 MIIC Shuffle=0 Threshold=0

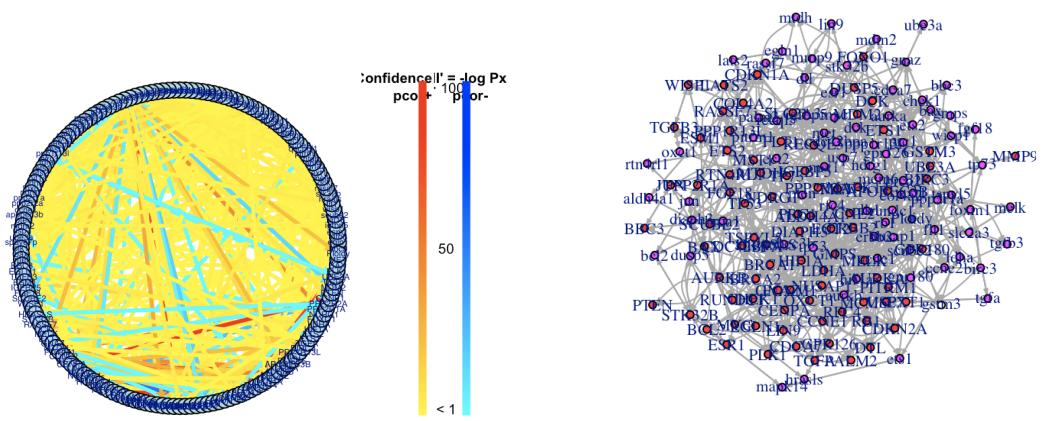


Figure 26: miic.plot et plot méthode MIIC 0 0

### 5.3.2 MIIC Shuffle=1 Threshold=0.001

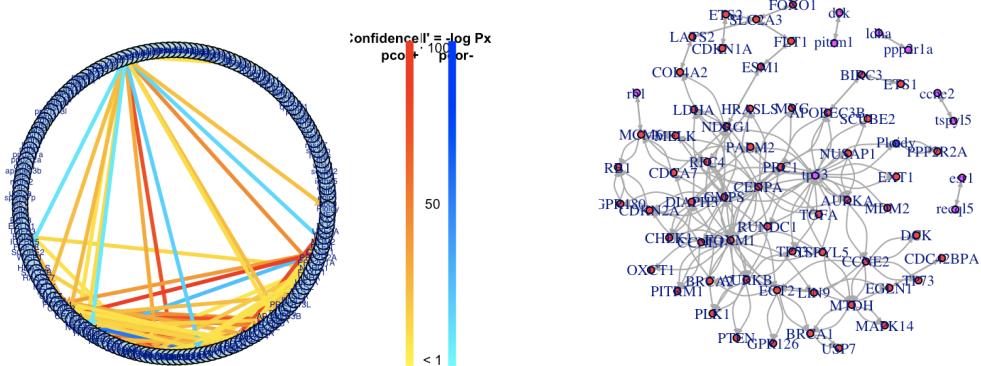


Figure 27: miic.plot et plot méthode MIIC 1 0.001

### 5.3.3 MIIC Shuffle=10000 Threshold=0.001

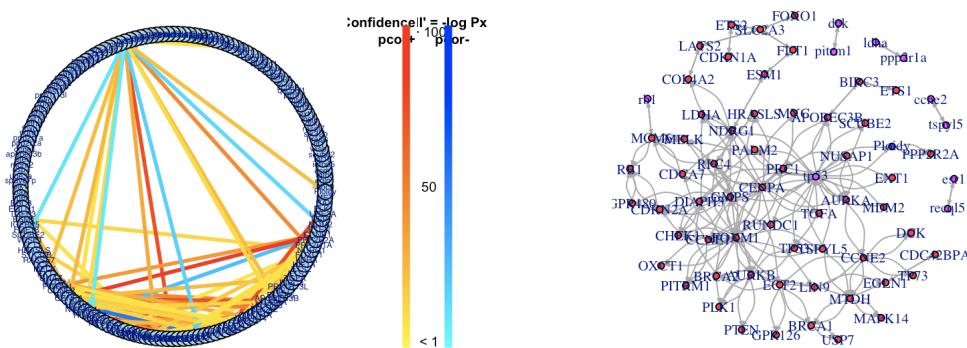


Figure 28: miic.plot et plot méthode MIIC 10000 0.001

#### 5.3.4 MIIC Shuffle=1 Threshold=0.00001

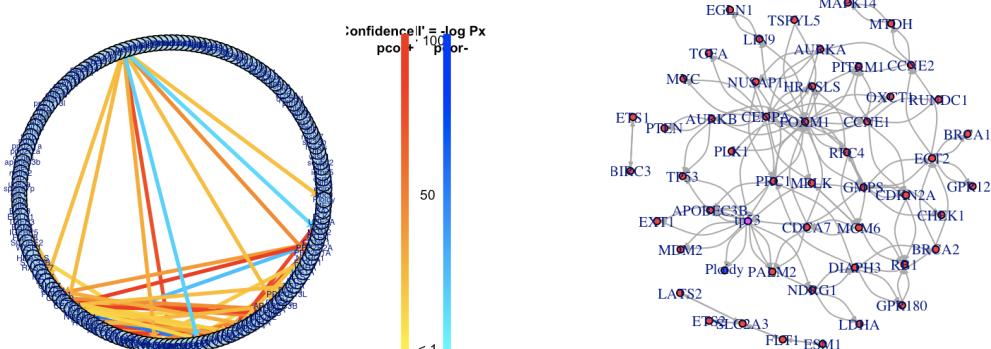


Figure 29: miic.plot et plot méthode MIIC 1 0.00001

Après avoir compris l'importance de ces hyperparamètres et leur influence sur la sortie de l'algorithme miic, nous nous sommes concentrés sur un graphe donc le nombre d'arrêtes n'était pas excédent, tout en conservant suffisamment d'information: shuffle 100, treshold 0.001.

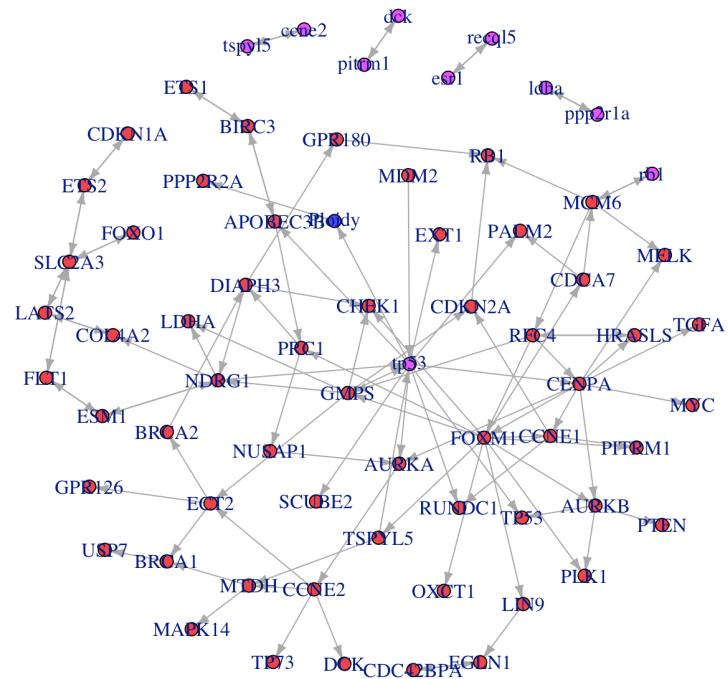


Figure 30: plot méthode MIIC 100 0.001

Nous avons également fait le choix de représenter ce graphe sous une forme hiérarchique afin de mieux comprendre les interactions et l'ordre de celles-ci:

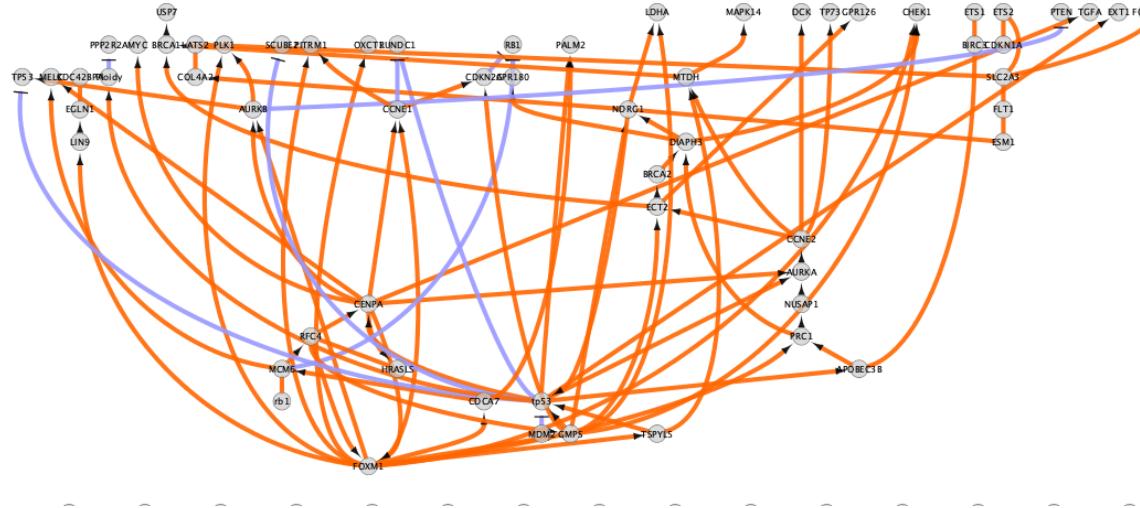


Figure 31: plot hiérarchique méthode MIIC

La méthode de reconstruction de réseau miic nous permet de souligner l'association directe qu'il existe entre la triploïdie et les mutations de tp53. Ces résultats sont également rassurants de par le fait que l'on trouve une relation entre les mutations des gènes BRCA1/2 avec la probabilité de triploïdie.

BRCA2	MTDH	PRC1	COL4A2	CDKN1A	FOXO1	ETS2	SLC2A3	rb1
1.729734e-17	3.459468e-17	3.459468e-17	6.173132e-06	6.173132e-06	6.562277e-05	6.690553e-05	3.158023e-04	9.942111e-04
FLT1	ETS1	LATS2	HRASLS	CDKN2A	GPR180	BIRC3	ESM1	CDC42
1.119033e-03	1.658743e-03	2.171161e-03	2.204951e-03	5.108975e-03	5.108975e-03	1.117448e-02	1.519592e-02	2.543070e-02
MDM2	CCNE2	TSPYL5	NDRG1	MCM6	NUSAP1	APOBEC3B	AURKB	DIAPH3
2.648914e-02	2.932596e-02	3.112518e-02	5.382429e-02	5.639996e-02	6.819640e-02	8.485722e-02	1.275219e-01	1.469736e-01
CCNE1	RFC4	GMPS	tp53	CENPA	FOXM1			
1.508265e-01	2.136439e-01	2.339636e-01	2.802423e-01	5.408007e-01	1.000000e+00			

Figure 32: variables considérées comme hub du graphe: méthode MIIC

Les hubs, ou éléments centraux de la mutation des autres gènes via la méthode miic sont donc FOXM1, CENPA, tp53, GMPS, RFC4.

```

1.368781e-01 2.55517e-01 2.669528e-01 2.988553e-01
> E(net2) [ to("Ploidy") ]
+ 1/111 edge from fa5e4b3 (vertex names):
[1] tp53->Ploidy
> E(net2) [ from("Ploidy") ]
+ 1/111 edge from fa5e4b3 (vertex names):
[1] Ploidy->PPP2R2A

> E(net2) [ from("tp53") ]
+ 9/111 edges from fa5e4b3 (vertex names):
[1] tp53->SCUBE2  tp53->EXT1  tp53->CENPA  tp53->RUNDC1  tp53->PALM2  tp53->TP53  tp53->AURKA  tp53->AP0BEC3B
[9] tp53->Ploidy
> E(net2) [ to("tp53") ]
+ 4/111 edges from fa5e4b3 (vertex names):
[1] TSPYLS5->tp53 GMPS ->tp53 NDRG1 ->tp53 MDM2 ->tp53

```

Figure 33: variables directement liée à la ploïdie et à tp53: méthode MIIC

```

> get.edgelist(net2)[index,]
      [,1]      [,2]
[1,] "tp53"    "CENPA"
[2,] "ESM1"     "NDRG1"
[3,] "FLT1"     "ESM1"
[4,] "CENPA"    "FOXM1"
[5,] "DIAPH3"   "NDRG1"
[6,] "SLC2A3"   "FLT1"
[7,] "BRCA2"    "DIAPH3"
[8,] "AURKA"    "CCNE2"
[9,] "AP0BEC3B" "PRC1"
[10,] "NDRG1"   "tp53"

> V(net2)[index]
+ 10/69 vertices, named, from fa5e4b3:
[1] tp53  ESM1  FLT1  CCNE2  CENPA  DIAPH3  SLC2A3  FOXM1  AURKA  NDRG1

```

Figure 34: Arrêtes et noeuds considérées comme ayant la plus grande centralité intermédiaire: méthode MIIC

Grâce à ces différents indicateurs, nous remarquons que la méthode MIIC arrive à très bien capturer l'importance de tp53: celui-ci étant présent à la fois comme hub, mais également comme faisant partie des noeuds et des arrêtes ayant la plus grande centralité intermédiaire.

## 6 Conclusion

Dans ce rapport, nous avons eu l'occasion de reconstruire un réseau de gênes et de l'analyser afin de mieux comprendre l'impact de certaines mutations des facteurs de transcriptions dans l'expression du cancer du sein et de l'impact de ces mutations sur la ploïdie des cellules tumorales.

Après une brève mise au point sur des concepts biologiques nécessaires à la bonne compréhension du sujet comme le principe de division cellulaire ou des mécanismes de contrôle liés à celle-ci, nous avons brièvement présenté les différentes approches des algorithmes de construction de graphe: basées sur l'optimisation de contraintes, de score, ou hybrides. Puis, nous nous sommes intéressés au jeu de données cosmic-Cancer, grâce à une rapide analyse de celui-ci, nous avons pu le préparer pour appliquer trois algorithmes choisis préalablement représentant chacune des approches existantes pour construire nos graphes.

Des trois graphes générés, nous nous sommes intéressés à plusieurs paramètres comme les gênes considérés comme des hubs, ou encore les gênes et relations considérées comme disposant de la plus grande centralité intermédiaire, ces paramètres nous auront permis d'avoir un retour de cohérence de nos graphes grâce aux différents papiers du domaine spécifiant certaines relations avérées. Ces paramètres nous auront également permis de comparer les uns aux autres afin de voir si nos résultats basés sur différentes approches étaient foncièrement similaires.

## References

- [1] Nadir Sella, Louis Verny, Severine Affeldt, Hervé Isambert *Learning Causal or Non-Causal Graphical Models Using Information Theory.*
- [2] Michael Sulak, Lindsey Fong, Katelyn Mika, Sravanthi Chigurupati, Lisa Yon, Nigel P Mongan, Richard D Emes, and Vincent J Lynch1 *TP53 copy number expansion is associated with the evolution of increased body size and an enhanced DNA damage response in elephants.*
- [3] Markus Kalisch, Peter Bühlmann *Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm.*