

UNIVERSITÁ DEGLI STUDI DI MILANO BICOCCA

Dipartimento di Informatica Sistemistica e Comunicazione



DATA ANALYTICS:  
Gene Regulatory Network of Mouse

**Docenti:**

Elisabetta Fersini  
Alex Graudenzi

**Autori:**

Simone Benitozzi 889407  
Christian Squadrito 889732

<b>Introduzione</b>	<b>4</b>
Obiettivi	5
Limitazioni dell'analisi	5
Software Utilizzati	5
<b>Network Analysis</b>	<b>6</b>
Analisi di Nodi e Archi	6
Degree Distribution	10
Shortest Path	12
Clustering Coefficient	13
<b>Centrality</b>	<b>15</b>
Degree Centrality	15
Betweenness Centrality	18
Closeness Centrality	20
Eigenvector e Pagerank Centrality	21
Reciprocity e Density	23
<b>Giant Component</b>	<b>25</b>
Degree Distribution	27
Shortest Path	28
Clustering Coefficient	29
Misure di Centrality	31
Degree	31
Betweenness	32
Closeness	32
Eigenvector e Pagerank	33
Reciprocity e Density	34
<b>Assortativity</b>	<b>35</b>
Analisi Assortativity	35
Resilienza	35
<b>Community Detection</b>	<b>49</b>
Algoritmi Testati	49
Algoritmi Cytoscape	49
Algoritmo Ottimale	56
Tuning Parametri OSLOM	59
Overlap nodi	62
Estrazione Community	67
Analisi Community	72
<b>Proprietà dinamiche Community</b>	<b>89</b>
<b>Conclusioni</b>	<b>91</b>
<b>Riferimenti</b>	<b>92</b>

# Introduzione

L'elaborato esporrà l'analisi di una Gene Regulatory Network di un *Mus Musculus* (topo comune).

Tale rete è un caso di studio interessante in quanto offre spazio a molte analisi compatibili con l'organismo umano perché è possibile prevedere malformazioni genetiche che possono portare allo sviluppo di cancri e altre patologie.

Questa rete modella la regolazione dell'**espressione genica**:

- I *geni* nel DNA codificano per le proteine
- La *DNA polimerasi* trascrive l'mRNA a partire da DNA (**processo di trascrizione**)
- l'mRNA viene tradotto in proteina a livello dei ribosomi (**macchinario traduzionale**)
- Le proteine sono coinvolte nella *funzione della cellula*
- Esistono *Fattori di Trascrizione*, tra le varie proteine, che promuovono o inibiscono l'espressione di determinati geni (**regolazione**).
- A loro volta i TF possono essere regolati da altre proteine
- Il gene “**attivato**” dal TF verrà trascritto e tradotto in proteina

I nodi di questa rete saranno i geni e i TF e gli archi orientati dai TF ai gene o dai TF ad altri TF creano un relazione di regolazione, che porta alla attivazione o disattivazione di un determinato gene o TF. Per astrarre ulteriormente la rete si possono vedere i TF come a loro volta dei geni dato che i TF non sono altro che il prodotto dell'espressione di quel particolare gene da cui prendono il nome.

Il dataset utilizzato è stato estratto da *RegNetwork*, specializzato in reti di questo genere, aggregazione di 17 database da altrettante fonti per informazioni riguardo la regolazione genica. Nel nostro caso, per semplificare il dominio del problema abbiamo pensato di rimuovere manualmente i nodi microRNA, perché regolano la post-trascrizione sulla quale abbiamo preferito non concentrarci. A livello post-trascrizionale gli miRNA agiscono mediante il riconoscimento di specifici mRNA targets al fine di determinare la degradazione o la repressione della traduzione e questo si manifesta nell'inibizione di particolari TF.

## Obiettivi

L'obiettivo del progetto è quello di analizzare innanzitutto la rete nella sua interezza, allo scopo di trovare informazioni chiave sul funzionamento del processo di regolazione genica.

Sarà importante trovare geni che svolgono funzioni rilevanti, come funzionano le interazioni tra di loro, e soprattutto tra diverse categorie di geni, e poter infine interpretare i risultati dal punto di vista biologico.

Vorremo anche sottoporre la rete al verificarsi di perturbazioni esterne e interne, per analizzare il comportamento dinamico e la resilienza della rete stessa.

Punto focale dell'analisi sarà poi il raggruppamento dei geni in cluster, con l'obiettivo di analizzare il comportamento di ciascuno di essi, estrarne un significato concreto e individuarne l'importanza a livello di funzionalità offerte all'organismo.

## Limitazioni dell'analisi

Una prima limitazione di fronte alla quale ci aspettiamo di scontrarci è la conoscenza non approfondita del dominio di riferimento e il campo biologico in generale. L'obiettivo sarà in ogni caso quello di effettuare analisi e ipotesi a partire dalle conoscenze tecniche a disposizione, per poi tradurre i risultati in qualcosa di concreto.

Un'altra limitazione è quella che riguarda la potenza di calcolo a disposizione. Trattandosi di una rete relativamente grande, e un caso di studio non banale, dal momento che le reti di regolazione dei topi sono a tratti molto simili a quelle degli umani, non tutte le analisi che avremmo voluto effettuare sono state possibili, come si vedrà in seguito, specialmente sulla parte di *community detection*.

## Software Utilizzati

L'analisi è stata portata avanti principalmente su 2 strumenti di sviluppo: *Python*, che in quanto linguaggio di programmazione ci ha garantito maggiore potere espressivo ed analisi più personalizzate.

La libreria maggiormente utilizzata in questo caso è stata *igraph*, specializzata nello studio di grafi e le proprietà.

In contrasto *Cytoscape*, specializzato nel dominio in questione e nella visualizzazione di reti biologiche, ci ha permesso di trattare in maniera più ottimizzata operazioni che in Python non erano eseguibili per la potenza di calcolo a disposizione, tra cui visualizzazione e community detection.

In aggiunta Cytoscape mette a disposizione plugin che ne aumentano le funzionalità base, e ciò ci ha permesso di sfruttare una funzione di *functional enrichment* per associare gruppi funzionali alle community trovate.

I plugin utilizzati sono stati *GeneMania*, utilizzato per strutturare il layout delle reti in fase di visualizzazione, facendo uso di assunzioni sul dominio specifico, e *CyCommunityDetection*, per gli algoritmi di community detection e l'arricchimento funzionale.

## Network Analysis

### Analisi di Nodi e Archi

Come visto in precedenza, la rete è costituita da due categorie principali di geni: i *target-gene* e i *Transcription Factor (TF)*.

Dal punto di vista tecnico, la differenza tra i due tipi di nodi è visibile rispetto alla direzione degli archi: dei 17.644 geni totali, ce ne sono solo 1.318 con connessioni uscenti, e consistono nei **TF**, mentre gli altri 16.326, aventi solamente connessioni entranti, quindi i nodi pozzo, rappresentano i **target-gene** di una rete di regolazione genica. In totale avremo 95.004 archi orientati.

I TF possono in ogni caso regolare tra di loro, e legarsi quindi, oltre che ai geni target, anche ad altri TF. Nel caso di una coppia di TF che si regola a vicenda, si parla di regulatory circuits, o feedback loop, e verranno analizzati successivamente nel dettaglio. Visto che una proteina svolge molteplici funzioni essa viene vista a livello di rete come un TF, ma nel frattempo al livello biologico potrebbe essere coinvolta non solo nella regolazione della trascrizione ma anche nella attivazione o disattivazione di un altro TF, ma nella semantica della rete stessa essa sarà vista sempre e solo come un semplice TF.

Gli archi della rete infatti rappresentano infatti la regolazione trascrizione che un gene effettua nei confronti di un altro gene, non vi è associato alcun peso e la rete non presenta multiarchi, per cui ogni connessione da un nodo all'altro è unica.

Da ciò deriva che solo i TF possono regolare l'espressione genica, e il comportamento dei loro archi uscenti ci permette di individuare due ulteriori sotto-categorie di geni: i *self-regulation genes* e gli *housekeeping genes*.

I *self-regulation genes*, 129 in totale, si contraddistinguono per il fatto di avere relazioni di self-loop, il che si traduce, nell'ambito del nostro dominio di applicazione, nell'autoregolazione del gene stesso tramite il suo TF associato. Questi tipi di geni sono fondamentali in quanto garantiscono alla rete di una maggiore robustezza e consistenza, in quanto la capacità di autoregolarsi li rende in grado di resistere al danneggiamento temporaneo di altri geni.

Ci sono in oltre 16 nodi TF che non presentano archi entranti e questa caratteristica può avere diverse interpretazioni: potrebbe trattarsi infatti di geni recettori dall'esterno, attraverso relazioni non rappresentate dalla nostra rete, oppure *housekeeping genes*, o geni costitutivi: si tratta di geni che vengono attivamente trascritti senza la regolazione di nessun TF. Generalmente, essi codificano proteine ed enzimi fondamentali per la vita della cellula, e che pertanto devono essere sempre presenti.

Attraverso una ricerca approfondita siamo arrivati alla conclusione che sicuramente alcuni di questi geni presentano funzioni di housekeeping, tra cui *Mphosph8*, fondamentale per sostenere l'auto-rinnovamento delle cellule staminali pluripotenti, e *Asb9*, che mostra una forte espressione in tessuti dei reni, cuore e muscoli.

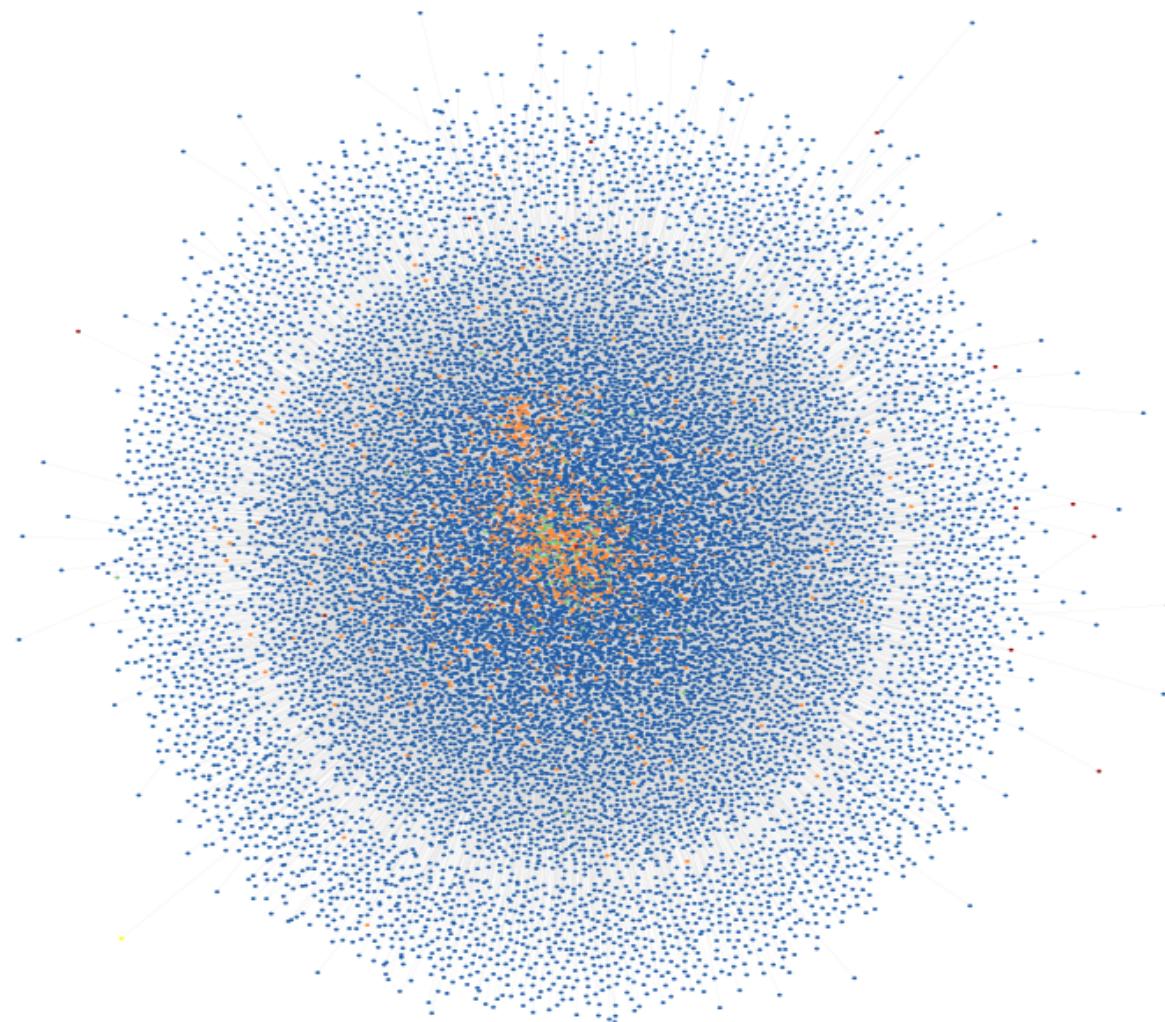
Non sono state trovate abbastanza informazioni per tutti i geni per assicurarci che tutti quelli con in-degree pari a 0 siano costitutivi, ma per raggrupparli tutti all'interno di un insieme comune, in maniera tale da poter effettuare analisi approfondite su di essi, d'ora in avanti assumeremo che siano tutti housekeeping.

In conclusione si può aggiungere che nessun TF è contemporaneamente un self-regulation e housekeeping gene, e tutti gli housekeeping genes hanno almeno un nodo in uscita, quindi rientrano a tutti gli effetti nell'insieme dei TF.

La seguente tabella riassume la distribuzione dei geni all'interno della rete, sulla base della loro categoria di appartenenza.

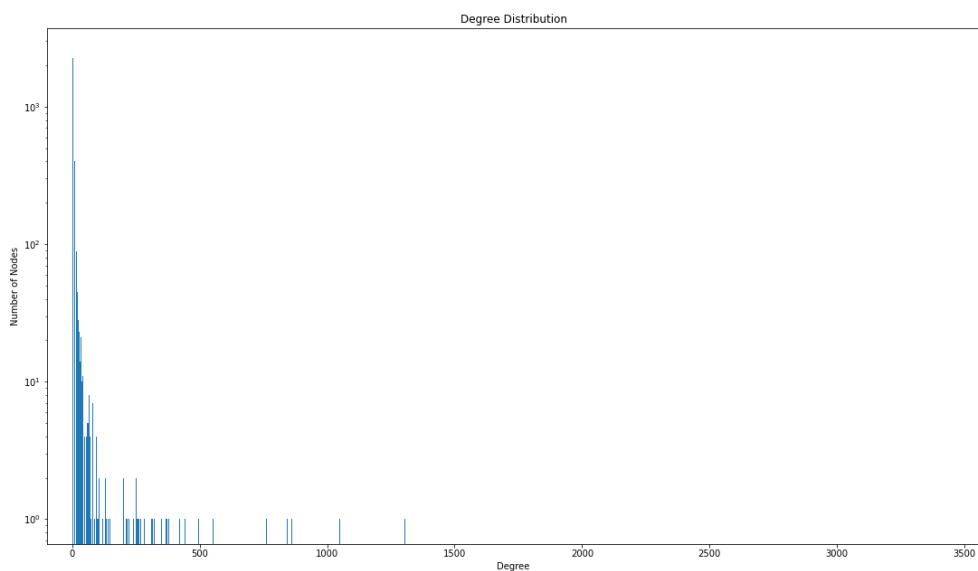
Suddivisione dei Nodi nella Rete	
CATEGORIA DI GENI	CARDINALITÀ
Totale	17.644
target-genes	16.326
Transcription Factors	1.318
TF - self-regulation	129
TF - housekeeping	16

La seguente rappresentazione della rete mostra la suddivisione dei nodi descritta finora, dove in blu sono rappresentati i target-gene, che costituiscono la maggior parte della rete e in arancione i semplici TF, presenti prevalentemente al centro. I TF di self-regulation, in verde, si trovano anch'essi verso il centro della rete, mentre gli housekeeping, in rosso, sono presenti anche in zone più periferiche.



## Degree Distribution

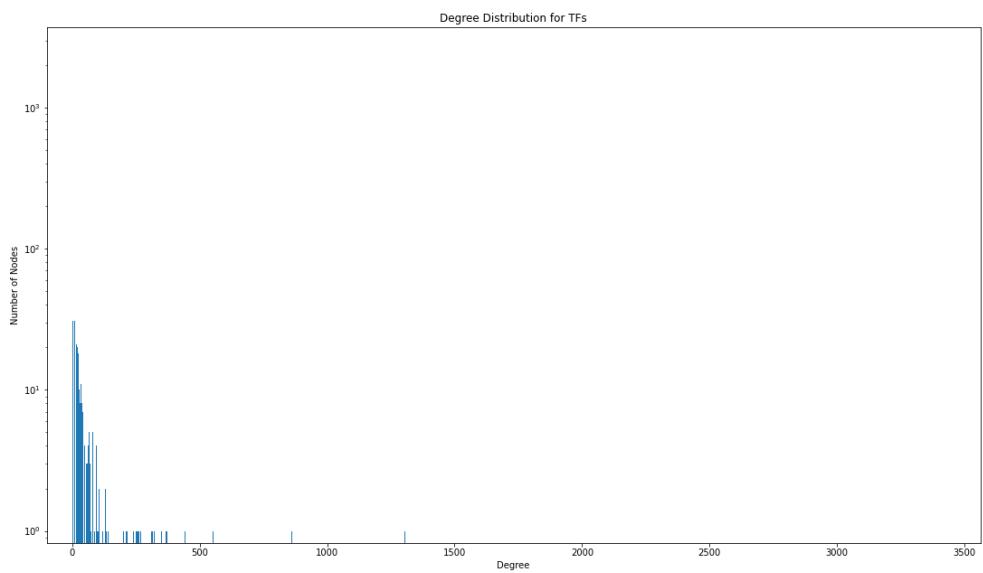
Quanto alla distribuzione di grado, la rete presenta una tendenza right-skewed, ed esponenziale negativa, che la rende di fatto **scale-free**, con la gran parte dei nodi con poche connessioni, e una frazione dei nodi con molti collegamenti, fino ad un massimo che supera le 3.000 connessioni. Questa distribuzione non sorprende, e anzi è molto tipica in reti di regolazione genica, caratterizzate da geni molto centrali che interagiscono con la maggior parte dei geni periferici.



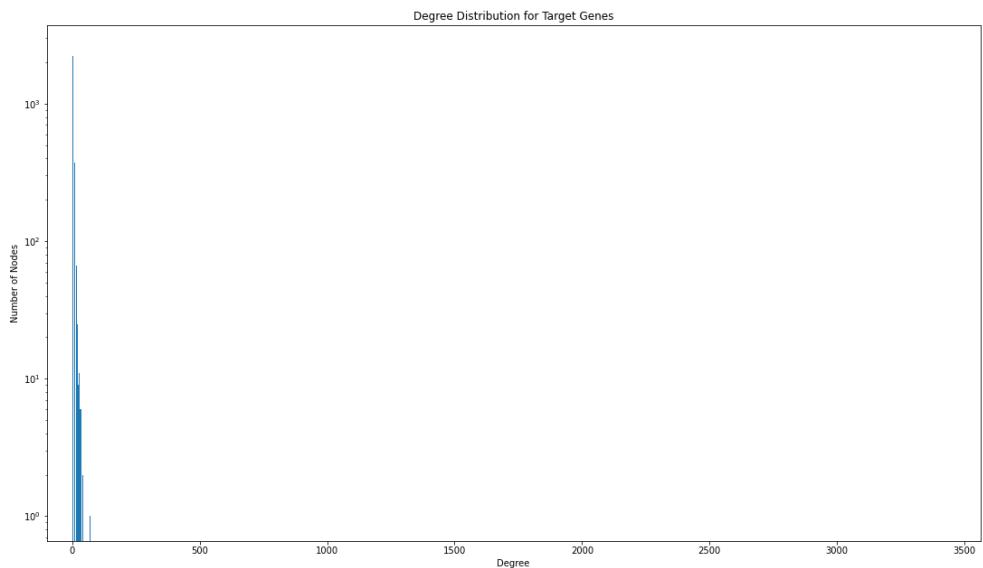
Questi risultati considerano tutti i nodi della rete alla pari, senza tener conto della categoria a cui appartengono. Pertanto in seguito l'analisi è stata approfondita in modo da separare i TF dai target-genes

Le distribuzioni di grado di entrambi presentano leggere differenze strutturali rispetto all'intera rete, pur avendo una forma simile alla distribuzione di grado totale. In entrambi i grafici si evidenzia una distribuzione di tipo right-skewed.

I TF, che definivano l'andamento della distribuzione di grado al crescere del degree, hanno un picco di ordinate molto più basso dei geni target, in quanto sono di numero molto ridotto, ma ci sono diversi nodi che superano il 100 di degree, arrivando fino ad un massimo di 3463 del TF *Ctcf*.



I geni target, che invece definivano l'andamento della distribuzione di grado al crescere delle ordinate, essendo di più, sono invece molto più schiacciati verso bassi livelli di degree, poiché, avendo solo archi entranti, non presentano molte connessioni quante i TF più connessi. Il massimo livello di degree in questo caso è 77, del gene *Ubc*.



## Shortest Path

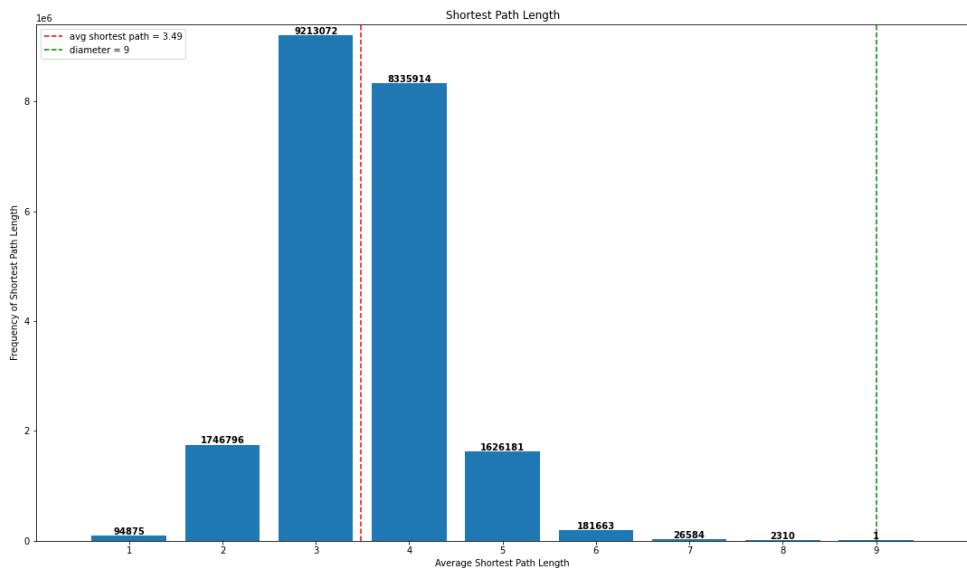
La rete ha uno shortest path medio di 3.49, con un comportamento da rete **small-world**, che è un valore abbastanza basso per un grafo di più di 17,000 nodi. Questo può essere attribuibile al fatto che gran parte dei collegamenti sono diretti da TF a target, e per il funzionamento della cellula le relazioni di regolazioni non possono essere troppo “lunghe” perchè l'espressione genica dovrà avvenire nel minor tempo possibile.

Il diametro è di lunghezza pari a 9, e c'è un solo shortest path di tale dimensione, che pertanto è stato possibile analizzare nel dettaglio. È costituito dai seguenti geni:

```
['Zfp524', 'Zfp641', 'Barx1', 'Dlx1', 'Sp7', 'Hhex', 'Sox8', 'Sp1',  
'Pttg1', 'Anapc4']
```

I 2 nodi di partenza, Zfp524 e Zfp641, che stanno per *zinc finger protein*, si occupano della codifica proteica. Il nodo più centrale è Sp1, con un degree di 262, mentre nessuno degli altri supera i 32 (Sox8).

Tutti i geni del cammino sono normali TF, tranne l'ultimo, *Anapc4*, che è un target-gene.



## Clustering Coefficient

Il coefficiente di cluster medio della rete è pari a 0.296, valore relativamente alto che si traduce nel fatto che i vicini hanno una buona probabilità di connettersi tra loro e formare una componente fortemente connessa.

Average Clustering Coefficient: 0.2958433239699835

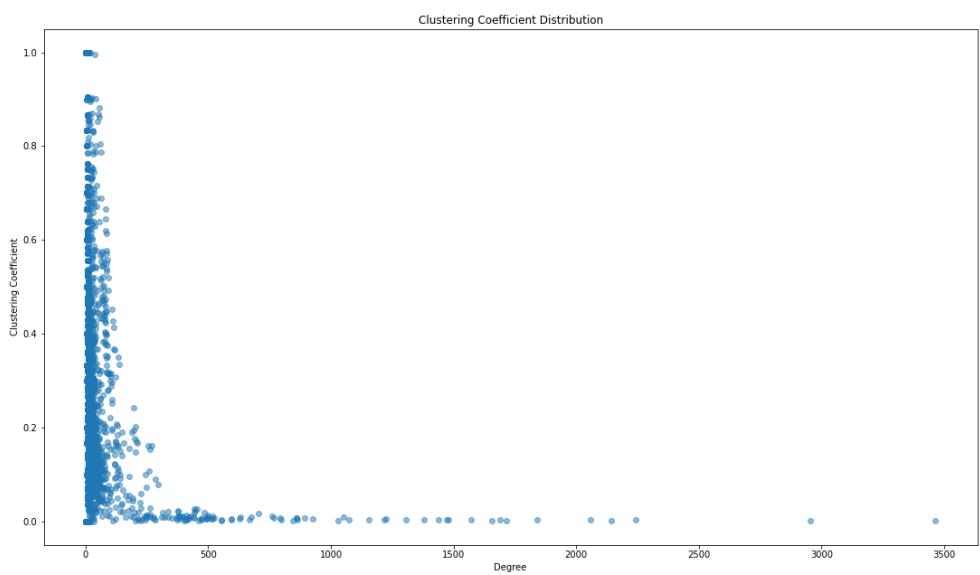
Il grafo si presta quindi bene al clustering dei suoi nodi, e questa è una caratteristica comune delle reti di regolazione, dal momento che i geni tendono effettivamente a raggrupparsi per svolgere funzioni comuni.

Come evidente dallo scatter-plot che mette in relazione degree e coefficienti di clustering, i nodi con coefficiente più alto sono quelli con degree molto basso, tendente all'1.

Al contrario gli hub della rete, con centralità di grado molto alta, tendono ad avere un coefficiente di clustering vicino allo 0, il che non sorprende, dal momento che sarebbe quasi impossibile che i vicini di un hub, che arrivano fino ad oltre 3000, possano solo accennare alla formazione di una componente fortemente connessa. Se a questo aggiungiamo che gli hub sono TF, e la maggior parte delle loro connessioni sono in direzione di target-gene, e quindi nodi pozzo, il risultato è prevedibile.

La rete presenta ben 1156 nodi con clustering coefficient pari esattamente a 1, i cui vicini costituiscono quindi una componente fortemente connessa. Di questi 1150 sono target-gene.

In quanto agli altri 6, si tratta di normali TF, tra i quali quello con centralità di grado maggiore, pari a 6, è *Taf1a*, responsabile della trascrizione dell'RNA polimerasi.



# Centrality

Dopo delle analisi preliminari sulla rete presa nella sua totalità, andremo ad analizzare più nel dettaglio il comportamento dei singoli nodi. Mostreremo solamente per la Degree Centrality, che l'utilizzo del calcolo della media per ogni centralità non è sufficiente per trarre conclusioni significative sulla rete; invece dovremo utilizzare la *centralization*, utile nel caso di grafi reali dove avremo valori di centralità molto variabili che rendono impreciso il calcolo attraverso la media della centralità.

## Degree Centrality

Dopo aver effettuato la distribuzione di grado sarà utile effettuare un'analisi puntuale sul grado per ogni nodo utilizzando la misura di centralità dell' in-degree, out-degree ed infine considerando la degree complessiva.

Partendo dall'in-degree e out-degree medio avremo questi risultati:

Average in-degree: 5.3844933121741105  
Median in-degree: 3.0  
Standard deviation in-degree: 6.664448409659355

Average out-degree: 5.3844933121741105  
Median out-degree: 0.0  
Standard deviation out-degree: 68.82208410639434

In media abbiamo 5 archi entranti e uscenti per ogni nodo però non possiamo fidarci appieno di queste misure sia per il fatto che non corrispondono alla mediana ed anche perchè si ha un'alta deviazione standard, soprattutto per il caso dell'out-degree. Anche nel caso della degree media avremo una situazione simile:

Average degree: 10.768986624348221  
Median degree: 4.0  
Standard deviation degree: 70.91490605813796

Si potrebbe sfruttare solo la deviazione standard per avere un'idea della variazione dei valori misurati, ma preferiamo affidarci ad una misura più di dominio come la Centralization:

in-Degree Centralization: 0.007687518044125775  
out-Degree Centralization: 0.19480842835472534  
Degree Centralization: 0.19558023871696892

Da questa misura abbiamo un'idea più precisa di come il grado dei nodi si comporta, ossia avremo che l'intera Degree Centralization è dominata dall'out degree dei nodi TF.

Per dimostrare ciò che è stato appena affermato, possiamo far vedere come quei nodi con un degree complessivo più alto corrispondono esattamente ai 5 nodi con l'out-degree più alto che sono appunto solo TF:

Top 5 Degree		Top 5 out-Degree	
Degree Centrality	Gene	out-degree Centrality	Gene
2059	Max	2028	Max
2144	Mycn	2130	Mycn
2244	E2f1	2215	E2f1
2956	Zfx	2940	Zfx
3463	Ctcf	3443	Ctcf

E' utile anche riportare i nodi con in-degree più alta che sono ancora una volta nodi TF dato che essi possono subire la regolazione da altri TF:

```

142    Pou5f1
101    Hdac1
100    Ep300
98     Foxp3
98     Esrrb

```

La forte interazione tra essi sono fondamentali per l'espressione genica ad esempio HDCA1 svolge un ruolo chiave nella regolazione dell'espressione genica eucariotica .

Invece se volessimo visualizzare quei nodi con più alta in-degree tra quelli target-gene troviamo:

```

77    Ubc
75    Ccnd1
70    Polr2l
55    Cdk7
53    Pcn

```

Essi hanno un coefficiente alto perché ricoprono un ruolo chiave all'interno della cellula e la loro trascrizione viene regolata da molti fattori di trascrizione, ad esempio *Ubc* è fondamentale nel mantenimento dei livelli di ubiquitina cellulare in condizioni di stress.

## Betweenness Centrality

Utilizzando la centralization possiamo notare come la misura di betweenness sull'intera rete sia molto alta, questo perchè ho molta variabilità nella misura vista la presenza di pochi nodi con un'altissima Betweenness Centrality e molti nodi pari a 0:

Betweenness Centralization: 93.94746818153729

Tra questi nodi con coefficiente di centralità molto alto troviamo solo nodi TF:

1660318.0814122881	Ctcf	TF self-regulation
1631000.3710590987	Pou5f1	TF self-regulation
1418802.9863210737	Esrrb	TF
1375312.9800686936	Tfcp211	TF self-regulation
1149297.0325163451	Tbp	TF
1020061.7725889218	E2f1	TF self-regulation
1003444.0815310596	Foxp3	TF self-regulation
990762.4866990958	Crebl	TF self-regulation
965579.6386651244	Myc	TF
881483.926275114	Max	TF self-regulation

Come ci si aspetta saranno presenti solo TF che fungeranno da ponte tra i gruppi di target-gene o ad eventuali altri gruppi di TF, dato che i nodi target-gene sono dei nodi pozzo in cui l'informazione si bloccherà. I TF si affermano come nodi centrali nel flusso informativo della rete e a livello biologico possono essere visti come dei TF che una volta regolati da un gruppo di TF a loro volta andranno a regolare un'altro gruppo di TF.

Possiamo anche far vedere che calcolando la edge betweenness otteniamo come uno dei due estremi i nodi con il coefficiente più alto:

419399.3185017253	Pou5f1	->	Ctcf
183366.41468908076	Tfcp211	->	Ctcf
146050.43869918896	Pou5f1	->	Crebl
142185.32345056295	Polrla	->	Ctcf
139821.8262738576	Pou5f1	->	Max
138178.6207879606	Ldb1	->	Gata1
125933.74179157593	Rorc	->	Ctcf
125627.12991210521	Foxp3	->	Ahr
116949.84604370687	Yy1	->	Zfx
115888.39370547776	Foxp3	->	Max

Come ci si aspettava l'arco che fungerà da ponte sarà anche quello che porterà un gruppo di nodi verso un nodo ponte, ma la cosa interessante è

vedere come i nodi ponte utilizzino gli stessi archi per interagire tra di loro come se cooperassero nel passaggio di informazioni. Si può vedere graficamente questa situazione in questa immagine:



Si può notare graficamente come i due TF utilizzino l'arco ponte per far fluire le informazioni recuperate da un nodo per passarle all'altro. Si può anche notare come il TF *Tfcp2l1* sia esso che principalmente regola questa dinamica nel garantire alla rete una stabilità ed una robustezza maggiore. Analizzando invece i nodi ponte si vedrà esplicitamente il fenomeno della numerosità degli archi entranti e di quelli uscenti che modella proprio ciò che è stato descritto a livello biologico.

## Closeness Centrality

Analizzando la misura di Centralization si nota come la variazione tra i vari coefficienti di Closeness è molto bassa:

Closeness Centralization: 4.797668518058744e-10

Ciò ci indica che tutti i nodi risultano posizionati centralmente rispetto all'epicentro della rete, anche perché la rete è risultata di tipo “small world” durante l'analisi del diametro e shortest path, quindi si hanno massimo 6 gradi di separabilità partendo da ogni nodo della rete. In questo caso non avremo solo nodi TF come nodi più centrali, ma anche nodi Target-gene anche se rappresenteranno dei nodi dove l'informazione convoglierà fermandosi.

Saranno quindi solo i TF con alta Closeness ad essere i responsabili di rendere efficiente e veloce l'inoltro di informazioni in tutta la rete, mentre i nodi target-gene saranno centrali dal punto di vista solo della ricezione delle informazioni perché avranno un'alta probabilità di essere raggiunti:

2.7696227773777213e-05	Ctcf TF self-regulation
2.744011195565678e-05	Zfx TF self-regulation
2.7216809101300964e-05	E2f1 TF self-regulation
2.6224005454593135e-05	Myc TF
2.580045924817462e-05	Cebpa TF
2.56910903298736e-05	Gata1 TF
2.559836170485089e-05	Creb1 TF self-regulation
2.5490046136983507e-05	Mycn TF
2.528189310815594e-05	<b>Ubc target</b>
2.5145213608589605e-05	Arnt TF
2.5097251850922323e-05	Jun TF self-regulation
2.508403150554357e-05	Esrra TF
2.506642602897679e-05	Klf4 TF
2.4996875390576178e-05	Tfcp2l1 TF self-regulation
2.49912530614285e-05	Pou2f1 TF self-regulation
2.497377753358973e-05	Mecom TF self-regulation
2.4951967462634428e-05	Max TF self-regulation
2.4951344877488898e-05	Yy1 TF self-regulation
2.4947610018960184e-05	<b>Mir92-1 target</b>

Quello che risulta interessante notare è che esistono nodi target centrali dal punto di vista della closeness e questo può essere spiegato dal fatto che, anche se i nodi target-gene sono definiti nella formulazione di questa Regulatory Network come dei nodi pozzo, in realtà i nodi target sono geni che a loro volta codificano dei nodi FT (quindi l'informazione in realtà passa) solo che questo aspetto non viene catturato dal modello perché si hanno

collegamenti in termini di regolazione dell'espressione genica dai TF ai geni, e *non di relazione di trascrizione dei geni verso i propri TF*.

## Eigenvector e Pagerank Centrality

Riportiamo i due valori di centralization:

Eigenvector Centralization: 5.500604445659014e-05

Pagerank Centralization: 7.270700007169674e-08

Sono due valori abbastanza bassi quindi possiamo dire che quasi tutti i nodi risultano avere importanza e rilevanza all'interno della cellula, ed infatti questo è ragionevole perché tutti i geni e TF sono utili (chi più chi meno) per il corretto funzionamento dell'organismo preso in esame. Sappiamo che in questa rete esistono dei nodi pozzo, ossia i nodi target, dove in realtà l'informazione continuerebbe a circolare ma per astrazione del modello stesso ciò non è presente.

Allora si può far vedere come esistano delle profonde differenze tra le due misure di centrality analizzando i nodi trovati con coefficiente più alto:

Eigenvector Centrality			Pagerank Centrality		
Centrality	Tipo	Gene	Centrality	Tipo	Gene
0.71547132 41331937	TF	Polr2i	0.00045024 4070633667 57	TF self	Mzf1
0.72149220 4669773	TF	Polr2j	0.00048104 8982971032 1	TF	Ssbp3
0.75121687 12329862	target	Cdk7	0.00048432 8273455817 9	TF	Esrrb
0.75161115 94592008	TF	Pparg	0.00048832 6544296854 3	TF self	Nanog
0.79178207 64738753	TF	Tbp	0.00051901 9898714038 1	TF self	Hdac1

0.79713102 83440721	TF	Med1	0.00052132 3147632730 2	TF self	Rnf2
0.89009400 00444143	TF	Polr2e	0.00053326 5599516170 1	TF self	Alx4
0.89858547 60397566	<b>target</b>	<b>Polr2l</b>	0.00058181 4289348749	TF	Tlx3
0.95289664 91670523	TF	Esrrb	0.00059740 6733229028 4	TF	Ldb1
1.0	TF	Polr2h	0.00095102 9560126314 8	TF self	Foxp3

Si possono già osservare due fenomeni:

- Il fatto che le misure di centralità sono più piccole per PageRank perché il coefficiente di influenza che un nodo conferisce ad un altro viene ripartito (diviso) tra tutti i nodi connessi tramite i suoi archi uscenti mentre eigenvector non tiene in considerazione questa assunzione così ottenendo una saturazione della misura più irrealistica .
- Nella misura di eigenvector appaiono nodi target che dovrebbero risultare i meno influenti rispetto ai TF che sono i principali regolatori dell'espressione genica. Questo può essere spiegato dal fatto che non vengono considerati casualmente possibili archi uscenti (come nel caso di PageRank) che potrebbero essere sfruttati per ritornare in alcuni nodi TF; così questi ultimi non riceveranno come contributo il coefficiente di centralità associato al nodo target (magari esso fa parte di quei nodi target influenti) e che potrebbe far "scalare" la classifica di quei nodi TF con più alto grado di centralità.

In generale si può dire che TF si confermano i nodi più rilevanti della rete dato il loro ruolo chiave nella regolazione genica. I nodi che spiccano sono quelli che si ripetono maggiormente con il prefisso di *Polr2* e sono relativi alla codifica e regolazione per la più grande subunità della RNA polimerasi II: l'ultima lettera dopo il prefisso corrisponde alla subunità di RNA su cui lavora.

## Reciprocity e Density

Infine analizziamo due misure che possono descrivere la rete rispetto alle misure di centralità prima riportate:

```
Reciprocity: 0.13069828722002635
# Feedback Loops: 6200
Density: 0.0003051914817306643
```

Come ci si poteva aspettare i valori di reciprocità e densità sono bassi, soprattutto quello della densità. Questo è da ricercare nel fatto che i TF si legano maggiormente con i nodi target-gene che non rispondono reciprocamente alla connessione dato che sono nodi pozzo. Inoltre la misura di reciprocità è strettamente legata alla densità perché senza un alto numero di archi di "risposta" tra i vari nodi, il grafo risulterà debolmente connesso. A livello biologico infatti i TF possono regolare i geni ma i geni non possono regolare a loro volta i TF ed infatti non c'è reciprocità tra di essi.

Si può effettuare un'analisi più approfondita e vedere quanti archi TF-TF abbiamo, dove ci aspettiamo un alto grado di reciprocità, e quanti archi TF-Target:

```
Archi TF-TF: 20084 (21.14%)
Archi TF-Target: 74920 (78.86%)
```

Possiamo dire di aver trovato diversi collegamenti tra TF e già avevamo intuito una qualche relazione tra di essi quando abbiamo trattato la Betweenness: a livello biologico questo si traduce in quello che è chiamato *feedback loop* o *regulatory circuits* che rendono più robusta la rete. Potremo vedere come cambia la misura di reciprocità e densità nei sottografi costruiti considerando solo archi TF-TF e un'altro sottografo con soli archi TF-Target:

```
Reciprocity TF-TF: 0.6213981458281133
Density TF-TF: 0.011802176621300802
```

```
Reciprocity TF-Target: 0.0
Density TF-Target: 0.000249330340703654
```

Come possiamo notare nel primo caso avremo un alto livello di reciprocità ma comunque non tutti i TF avranno collegamenti reciproci tra di loro; nell'altro caso avremo alcuna reciprocità tra i nodi, come già detto, ed anche una

bassissima densità che sarà quella che farà crollare il valore complessivo dell'intera rete.

## Giant Component

Un'altra analisi effettuata è stata quella della connettività del grafo. La rete risulta debolmente connessa, non ci sono quindi nodi o componenti isolate, e l'intera rete costituisce quindi la *Giant Component* debolmente connessa.

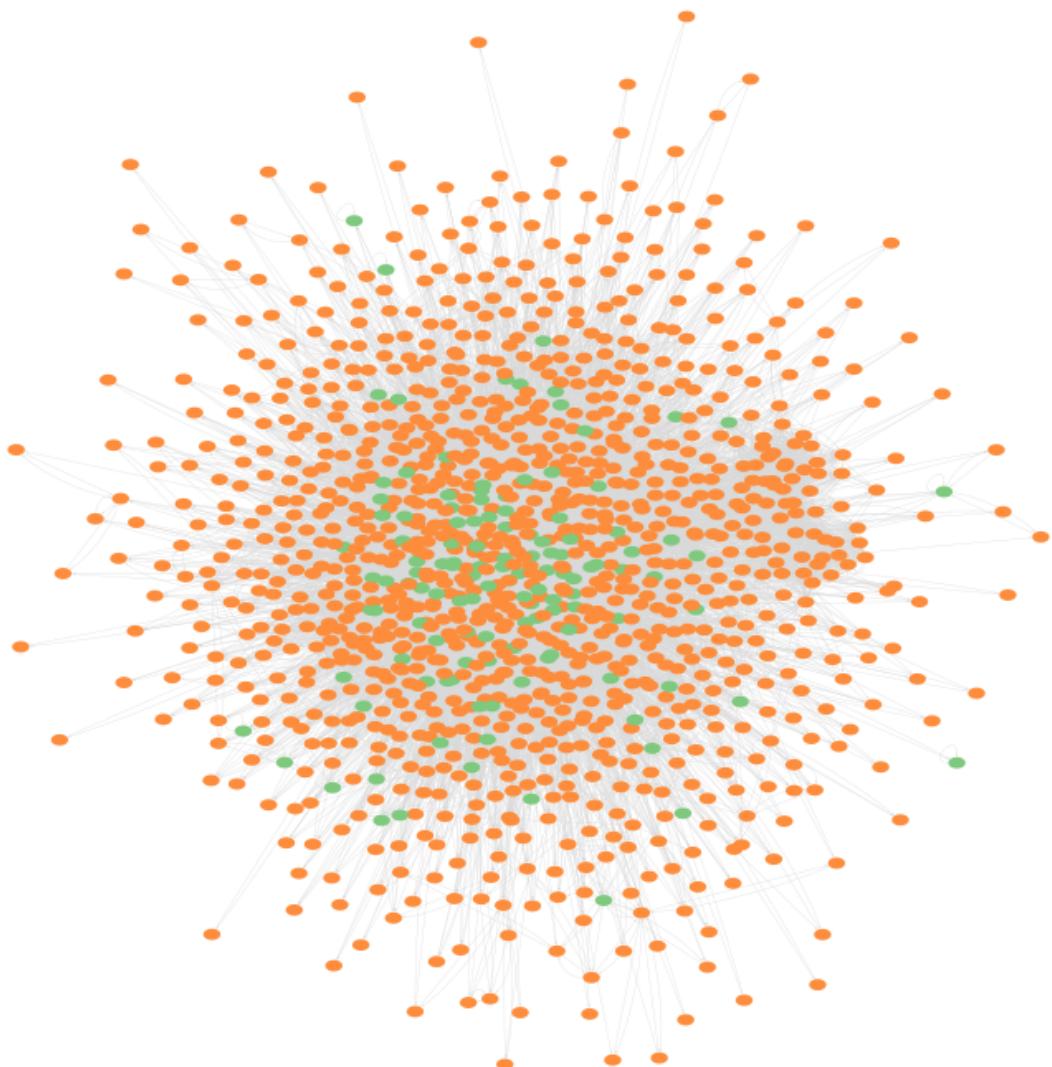
Quanto alla connessione forte, la rete non lo è, in quanto già solo la presenza di nodi pozzo, quali i target-gene, non permette che tale proprietà possa verificarsi.

Sono quindi state analizzate le componenti fortemente connesse: la più grande presenta 1203 nodi, seguita da due molto piccole, di rispettivamente 3 e 2 nodi. Per il resto, tutte le altre componenti isolate sono costituite da un singolo nodo.

D'ora in avanti definiremo quindi come Giant Component la componente fortemente connessa più grande. Essa presenta 1203 nodi, tutti TF, di cui 126 di self-regulation e nessun housekeeping.

Risulta interessante come dei 129 TF di self regulation, il 98% sia incluso nella Giant Component, con soli 3 mancanti. Questo conferma come tali geni costituiscano un ruolo fondamentale nella rete, come vedremo successivamente anche nell'analisi delle community in cui essi si presentano.

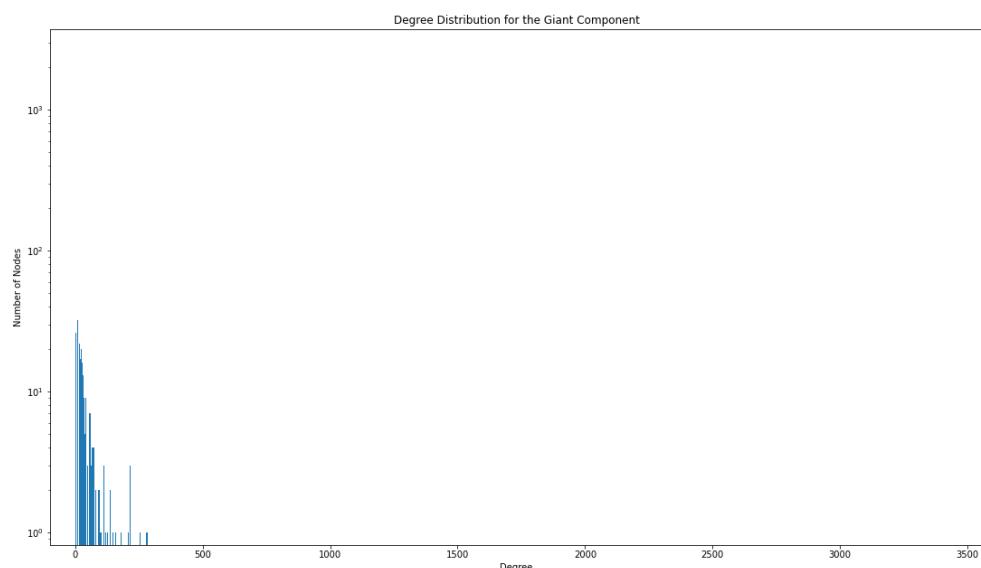
La Giant Component così ottenuta è rappresentata di seguito. Si ricorda che i nodi arancioni rappresentano i TF semplici, mentre i verdi i self-regulation TF, che si vede assumere un ruolo centrale all'interno della sottorete.



## Degree Distribution

La forma della distribuzione di grado rimane simile a quella dei TF della rete originale, ma con valori più bassi, dovuti all'abbassamento dell'out-degree, avendo rimosso tutte le connessioni in uscita verso i nodi target.

Il picco viene raggiunto, con degree centrality pari a 284, dal gene *Esrrb*: si tratta di un Transcription Factor fondamentale per la stabilizzazione del DNA. La sua importanza dal punto di vista pratico si riflette quindi nell'analisi della Giant Component, nella quale svolge un ruolo centrale.

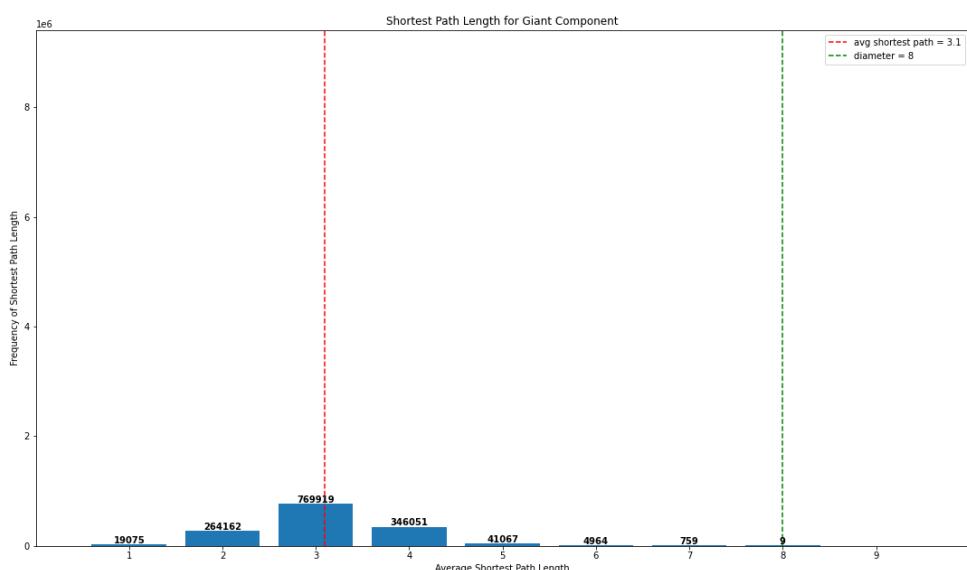


## Shortest Path

L'average shortest path della Giant Component è leggermente minore, 3.1 contro 3.49 della rete, e ci si poteva aspettare che sarebbe diminuito, sebbene non di molto, in quanto per ogni cammino viene tagliato almeno il nodo finale, rappresentato da un target gene

Il diametro questa volta è 8, e ci sono 9 cammini di tale lunghezza. Ci aspettavamo che sarebbe stato minore 9, in quanto il diametro nel grafo completo comprendeva un target-gene alla fine.

Dal grafico è evidente come il numero di cammini in ogni caso sia decisamente minore della rete originale, ma in proporzione la tendenza è più o meno la stessa, con un picco per i valori 3 e 4, seguiti dai due intorni 2 e 5

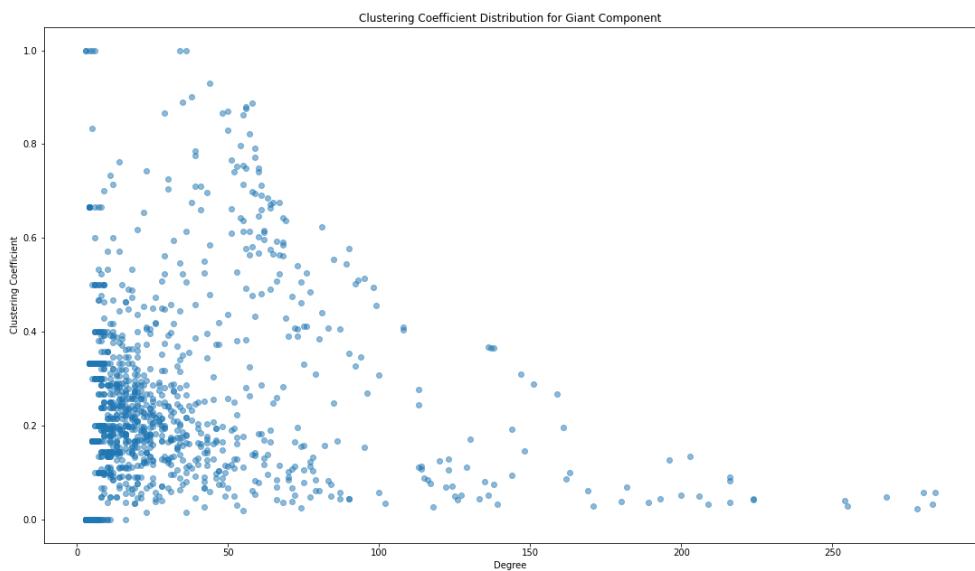


## Clustering Coefficient

La Giant Component presenta una tendenza più bassa al clustering rispetto all'intera rete (0.26), nonostante abbia molti meno nodi. Questo è probabilmente dovuto al fatto che i tanti target-genes con alto coefficiente, che qui scompaiono, tendevano ad avere un neighborhood che costituisse una rete completa. Inoltre nella rete originale i cluster che si formano sono costituiti da molti target genes periferici e pochi TF che li dominano, che è anche una situazione più "naturale", rispetto ad avere cluster di soli TF

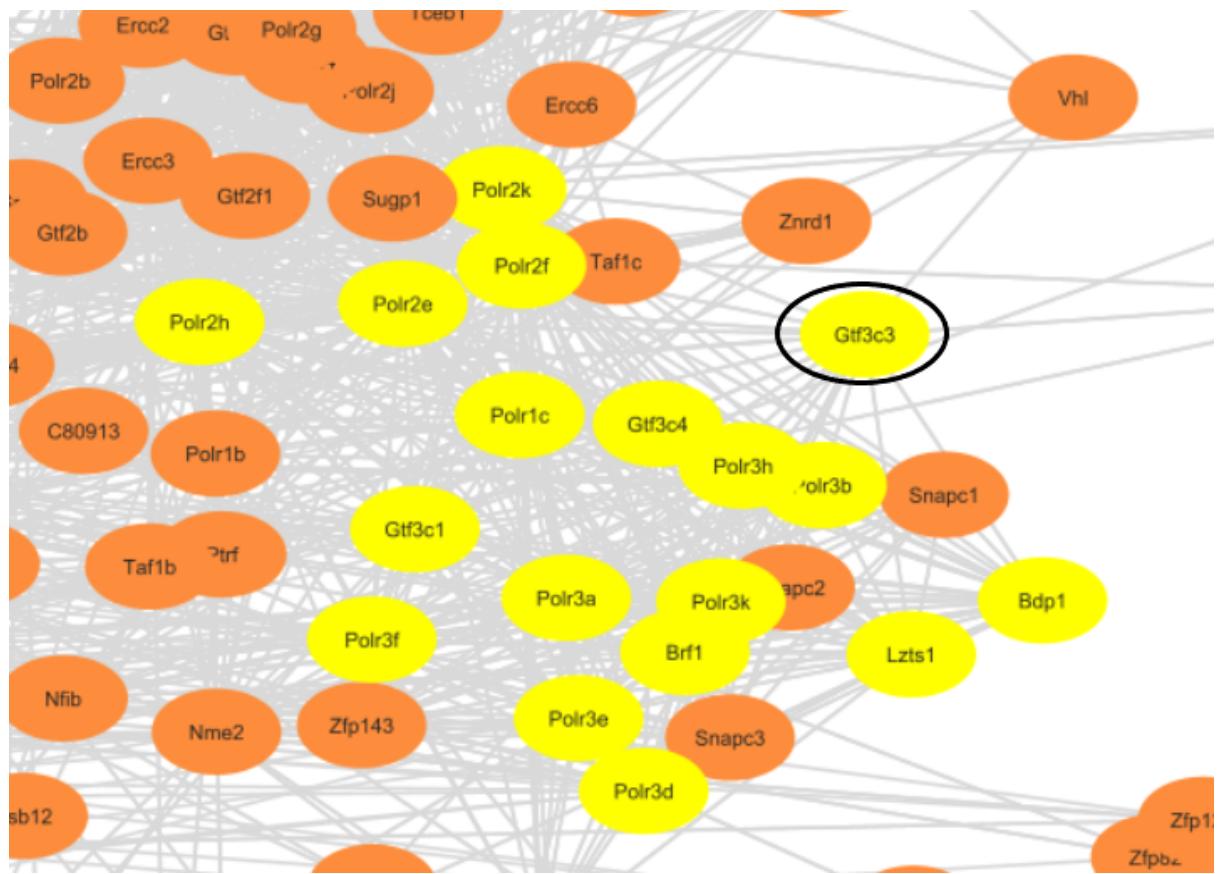
Average Clustering Coefficient: 0.24447507810223665

Lo scatter-plot mostra nette differenze rispetto alla rete originale. Abbiamo 7 nodi con coefficiente esattamente a 1.0, di cui nessun TF di self-clustering. Si tratta del 0.58% del totale, contro il 6.55% della rete originale.

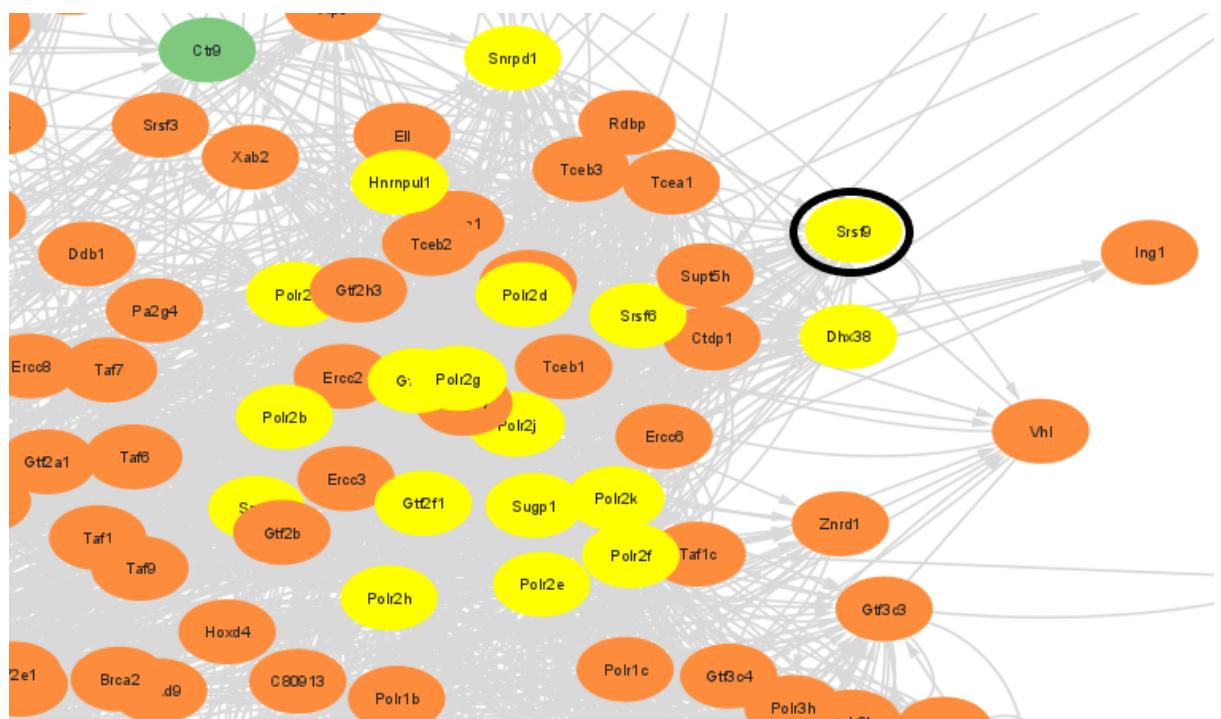


Rispetto a prima, qui troviamo nodi con coefficiente a 1.0 anche per degree più alti: *Gtf3c3* e *Srsf9* hanno rispettivamente 36 e 34, e i loro vicini formano comunque una rete fortemente connessa.

*Gtf3c3* è un gene coinvolto nella trascrizione dell'RNA polimerasi, difatti nella seguente porzione di Giant Component, in cui sono evidenziati anche tutti i suoi vicini di primo ordine, troviamo tutti i geni del tipo *Po2r\_*, che abbiamo già incontrato in precedenza.



*Srsf9* è un TF coinvolto invece nel processo di splicing dell'RNA messaggero, e si nota come condivida alcuni vicini proprio con *Gtf3c3*: i due potrebbe infatti essere coinvolti in funzioni condivise.



## Misure di Centrality

### Degree

Trattandosi di una sottorete contenente solo TF, l'analisi è stata fatta direttamente sulla degree distribution generale.

```
Average degree: 31.92186201163757  
Median degree: 18  
Standard deviation: 39.11802882166683
```

Rispetto alla rete, abbiamo un average degree 3 volte più alto, con 32 anziché 10, ma questo era preventivabile dal momento che spariscono i nodi pozzo che ne abbassavano molto la media.

Anche in questo caso media e mediana non corrispondono, e la deviazione standard risulta decisamente alta.

Ci sarà pertanto bisogno di analisi più dettagliate attraverso la distribuzione di grado.

```
Degree Centralization: 0.2100648239611749
```

La degree Centralization risulta più alta, 0.21 rispetto a 0.19, il che riflette il fatto che già nella rete essa era dominata quasi totalmente dai valori di out-degree, quindi dai TF. Essendo la Giant Component interamente costituita da TF, la tendenza ad aumentare, seppur di poco, era attesa.

I nodi con degree centrality maggiore risultano essere i seguenti:

```
284 Esrrb  
283 E2f1  
280 Pou5f1  
278 Zfx  
268 Tfcp211
```

Si ritrovano quindi *E2f1* e *Zfx*, che erano nella top 5 dell'intera rete, manca invece *Ctcf*, che era il nodo più centrale, probabilmente a causa di molte più connessioni con target-genes, che nella Giant Component spariscono.

## Betweenness

La Giant Component assume un comportamento simile a quello della rete in quanto a betweenness, con una centralization praticamente alla pari (93.9)

Betweenness Centralization: 94.6000913392098

Questo si traduce in una eguale capacità dei nodi più centrali di fare da ponte tra diversi geni della rete

Quanto ai nodi più centrali, anche in questo caso manca Ctcf, che nella rete presenta il valore più alto anche per la betweenness. Gli altri nodi tendono invece a seguire la tendenza già vista in precedenza.

116043.83961551626	Pou5f1
74477.51921762004	Foxp3
66847.6158679778	Tbp
66247.86595418482	Esrrb
63613.43650544507	Tfcp211

## Closeness

A livello di capacità dei nodi di diffondere informazioni, la Giant Component risulta essere più centralizzata rispetto alla rete complessiva (4.797e-10), il che si spiega con il fatto che avendo solo TF, c'è un maggior flusso di informazioni e una rete più fitta di connessioni

Closeness Centralization: 1.0181497680310489e-07

Per quanto riguarda i nodi più centrali, ritroviamo esattamente gli stessi geni, in questo caso con un fattore esponenziale di differenza (rispetto a ~5e-05), segno del fatto che quelli con un maggior capacità di inviare informazioni restano gli stessi, ma aumenta la loro efficienza nel farlo

4.5475216007e-04	Zfx
4.5167118338e-04	E2f1
4.4883303411e-04	Ctcf
4.3706293706e-04	Myc
4.3459365493e-04	Crebl

## Eigenvector e Pagerank

I valori di Centralization per le due misure risultano simili alla rete originale, con il solo aumento di una potenza di 10 (da e-08 a e-07) nel caso della Pagerank Centralization

Eigenvector Centralization: 5.057639820258274e-05

Pagerank Centralization: 6.359006876273423e-07

Anche in questo caso i geni con centralità più alte restano gli stessi della rete originale, e per Eigenvector Centrality ritroviamo in maggioranza quelli responsabili della codifica dell'RNA polimerasi.

Eigenvector Centrality			Pagerank Centrality		
Centrality	Tipo	Gene	Centrality	Tipo	Gene
0.99999999 99999999	TF	Polr2h	0.00598331 3876091079 5	TF self	Tcf3
0.95289664 91670539	TF	Esrrb	0.00619959 8375252899	TF self	Hdac1
0.89009400 00444155	TF	Polr2e	0.00653785 6246137948	TF	Nfe2
0.88240763 68913127	TF	Polr2k	0.00871476 2883263782	TF self	Foxp3
0.88082318 95631803	TF	Polr2f	0.01204048 9640239516	TF self	Pou5f1

## Reciprocity e Density

Come ci si poteva aspettare, entrambe le misure sono più alte della rete originale, in quanto nella Giant Component ci sono solo TF, che possono avere connessioni bidirezionali

Reciprocity: 0.6497509829619922

# Feedback Loops: 6197

Density: 0.013278644763576361

La reciprocity risulta infatti 5 volte più alta, con un valore molto elevato di 0.65. Questo ad attestare il fatto che la Giant Component sia ricca di *feedback loop*, solo 3 in meno della rete originale, che gli conferiscono una struttura molto robusta, dal momento che in 1 caso su 3, due TF si regolano a vicenda.

La density è invece addirittura 40 volte più alta dell'intero grafo, ma la sottorete risulta comunque non molto densa, con un valore di 0.013.

Come visto anche nel coefficiente di clustering, questo è dovuto al fatto che sebbene ci siano parti della rete che addirittura formano sottografi completi, allo stesso tempo ci sono altri nodi i cui vicini sono poco collegati tra loro, e questo contribuisce a diminuire l'intera misura di densità.

# Assortativity

## Analisi Assortativity

Effettuiamo un'analisi dell'assortatività per poter comprendere meglio la dinamica della rete riguardo alla correlazione di grado in modo da evidenziare quale sia il comportamento che sussiste tra hubs e spokes. Il valore rilevato sarà:

```
Network Assortativity: -0.30129304144666413  
Giant Component Assortativity: -0.22887240169742862
```

La rete come si può vedere è disassortativa quindi gli hubs tenderanno a legarsi con gli spokes e viceversa. Come è già stato detto più volte i nodi TF (Hubs) avranno un comportamento chiave all'interno della rete quindi saranno essi che si legheranno con i nodi gene-target (Spoke) per regolare la loro trascrizione. Come già visto anche gli Hubs (TF) si collegano con altri Hubs (TF), ma questo fenomeno non risulta così incisivo per rendere la rete assortiva e soprattutto perché non avremo nemmeno situazioni in cui spoke (gene-target) si legano con spoke (gene-target).

Si può anche notare che prendendo in esame esclusivamente la Giant Component molti nodi spokes non saranno presi in considerazione, e la rete risulta meno disassortitiva come si poteva immaginare.

Successivamente andremo ad analizzare l'assortatività in ogni community cercando di capire se la stessa dinamica si riscontra anche in partizioni della rete stessa.

## Resilienza

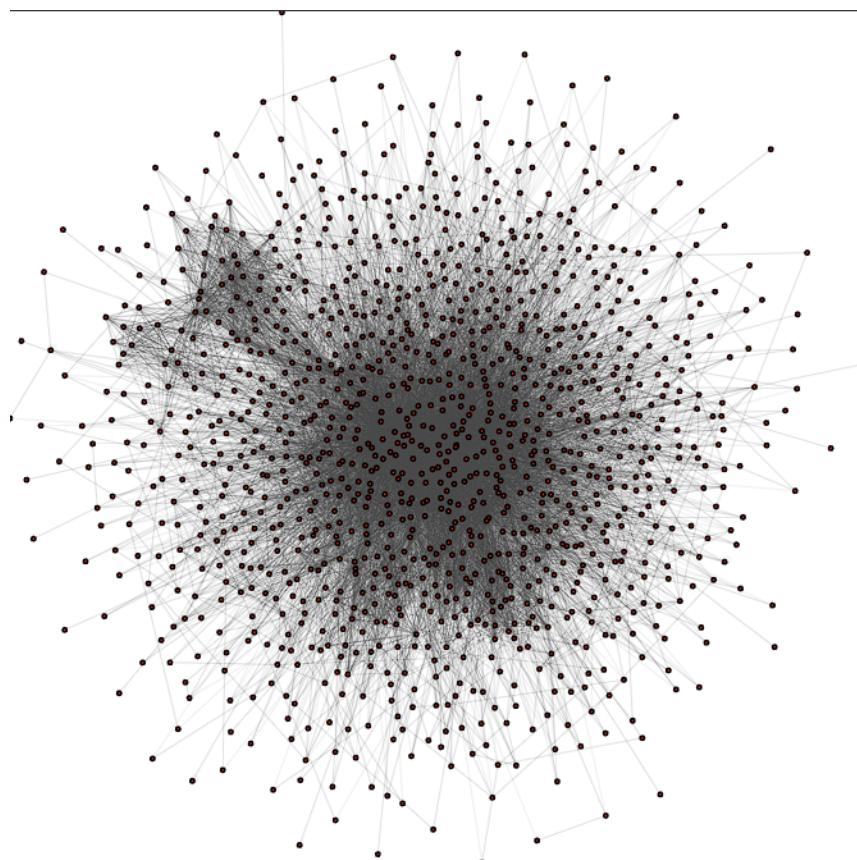
Ora cercheremo di verificare la robustezza della rete effettuando una rimozione di alcuni tipi di nodi, facendo vedere fino a che punto la Giant Component ed altre misure di Network Analysis resistono alle perturbazioni comparandoli alle misure prima eseguite sulla rete non perturbata.

Ci è sembrato inutile andare a rimuovere direttamente gli archi, perché sarebbe stato un processo più dispendioso rispetto alla semplice rimozione di nodi che già portano alla rimozione di parecchi di essi. Inoltre andremo a rimuovere nodi non in modo casuale, ma i nodi più importanti per poter verificare il comportamento della rete in modo più esplicito.

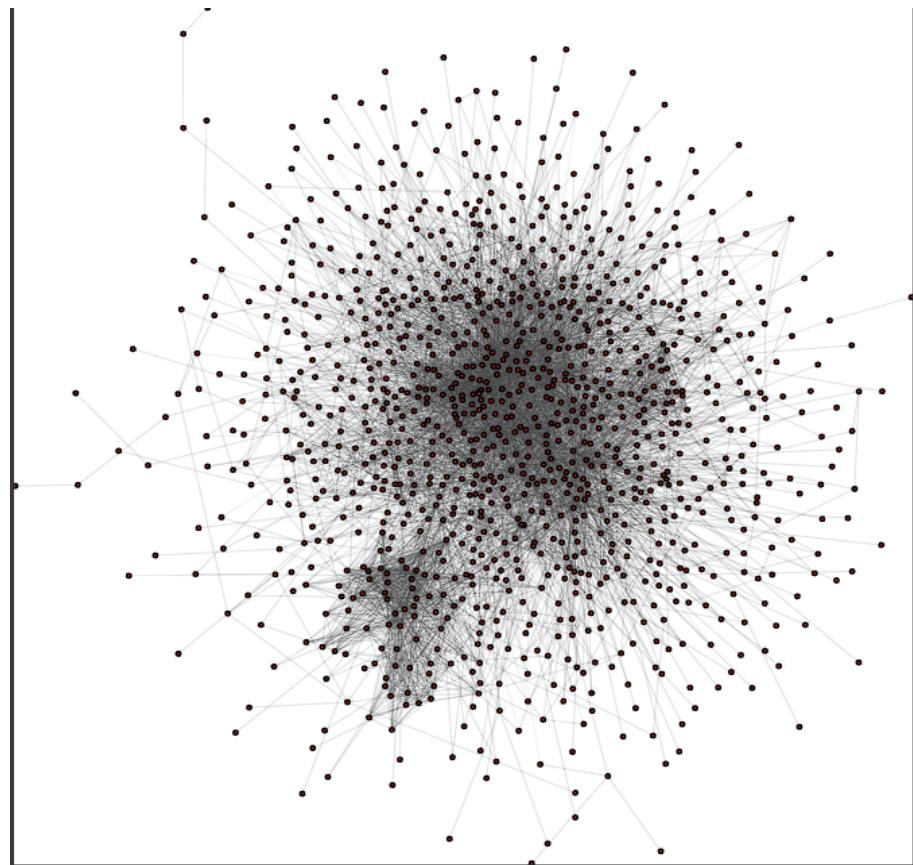
Andremo a verificare la resilienza prima eliminando gli hubs, poi testandola sui nodi brokers ed infine rispetto ai nodi con closeness più alta. A livello

biologico se un TF viene danneggiato questa potrà essere ritrascritto da un gene ma se una parte del DNA viene danneggiato in maniera irreparabile, allora tutta la funzionalità della cellula è compromessa. Per semplicità andremo ad eliminare anche i TF, così ottenendo una situazione momentanea della rete che poi sarebbe capace, a livello biologico, di recuperare la sua situazione iniziale. Con questa assunzione procediamo nella rimozione di entrambi i tipi di nodi astraendo questo comportamento biologico.

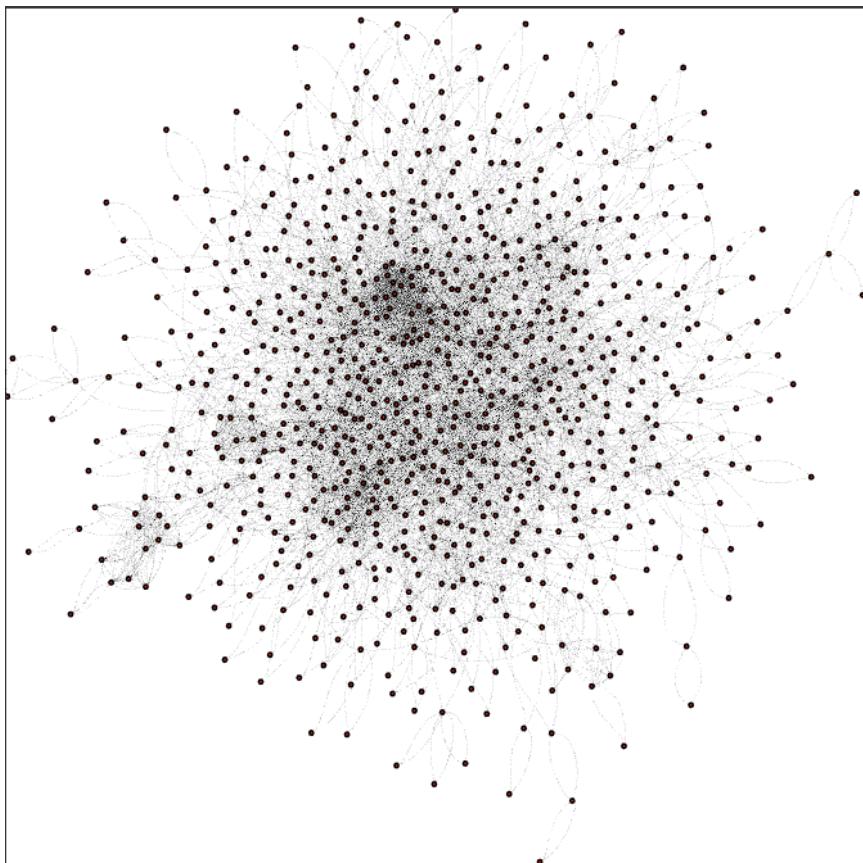
Di seguito è come si presenta la Giant Component senza alcun taglio:



Partiamo considerando la rimozione degli hubs:



Già si può notare come con la rimozione di soli 50 hubs la giant component si sia già ridotta considerevolmente. Se continuiamo con le rimozioni arriviamo ad una situazione con 200 hubs rimossi in cui la Giant Component scomparirà quasi completamente:



Misurando la densità, la reciprocità e Average Clustering Coefficient e comparandola con quella originaria ci si accorge:

Giant Component Before:

Reciprocity: 0.6497509829619922

Density: 0.013278644763576361

Average Clustering Coefficient: 0.2462836843281994

Giant Component After:

Reciprocity: 0.251286243208155

Density: 0.0001347295340222878

Average Clustering Coefficient: 0.17259521132446085

Anche attraverso misure più analitiche ci si accorge come la rete risulta essere meno connessa. Allora ci potremmo chiedere quanti nodi rimarrebbero isolati e quante componenti fortemente connesse avremo rispetto a prima:

Giant Component Before:

Nodi isolati: 0

strongly connected components 1

```
Giant Component After:  
Nodi isolati: 6387  
Nodi isolati TF: 6  
Nodi isolati Target: 6381  
strongly connected components 16474
```

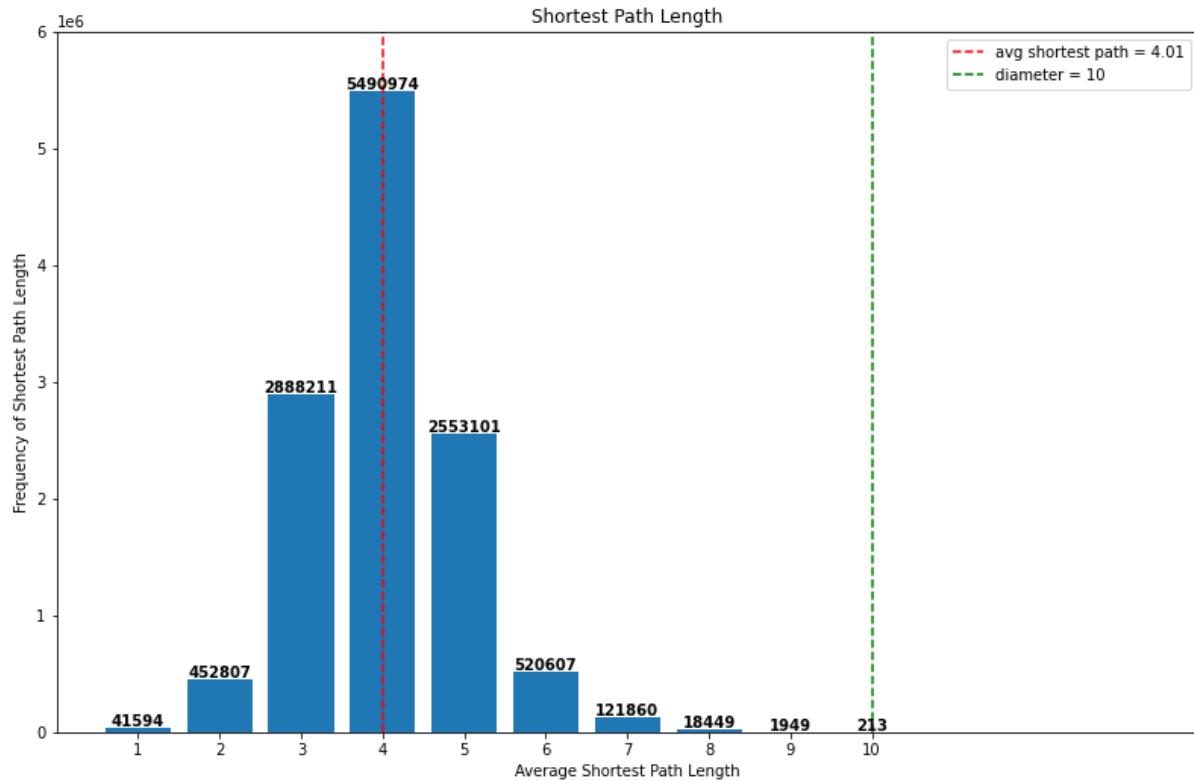
Prima della rimozione degli hubs non avevamo nodi isolati perché non possono esistere geni che non vengono regolati da nessun TF ed ha una componente fortemente connessa che è la Giant Component stessa.

Si può notare, dopo il processo di rimozione, la formazione di diversi nodi isolati e componenti fortemente connesse, traducendosi questo nel fatto che la rete rimossi 50 hubs si perderanno molte connessioni con quasi la metà dei geni-target che rimarranno completamente disconnessi dalla rete e ci saranno diverse componenti della rete che cercheranno di “sopravvivere” facendo sì che i nodi al loro interno cooperino tra loro come nel caso di nodi feedback loop. Inoltre tra i nodi isolati ci sono 6 nodi TF:

```
['Zfp523', 'Mxd4', 'Mxd3', 'Hopx', 'Btaf1', 'Batf3']
```

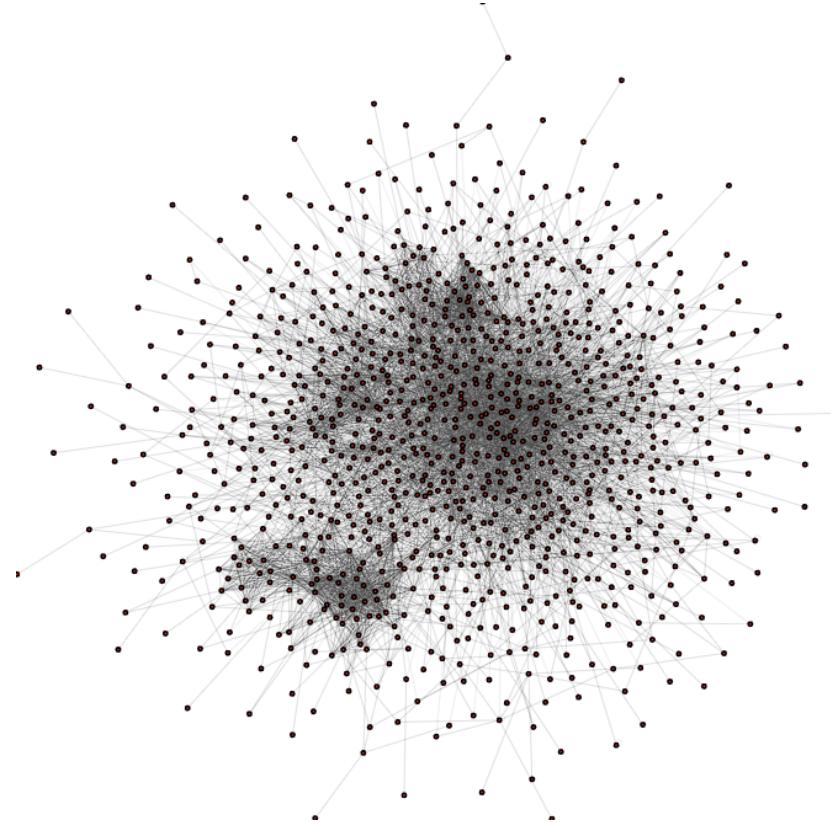
Si tratta di nodi TF con una bassa degree, ad esempio *Zfp523* fa parte della famiglia di proteine che permettono di legare il DNA ma risulta quella meno citata sui vari database biologici quindi presumo che sia la meno influente. Quindi danneggiando anche solo 6 di questi TF non risulta un grande danno per la cellula solo che il numero dei geni rimasti fuori dalla Giant Component è enorme e questo ci fa intuire che i TF riescono in qualche modo a trovare un compromesso aiutandosi tra loro ma i geni saranno sempre quelli più in difficoltà.

Ci si aspetta che anche lo Shortest Path e il diametro siano aumentati:

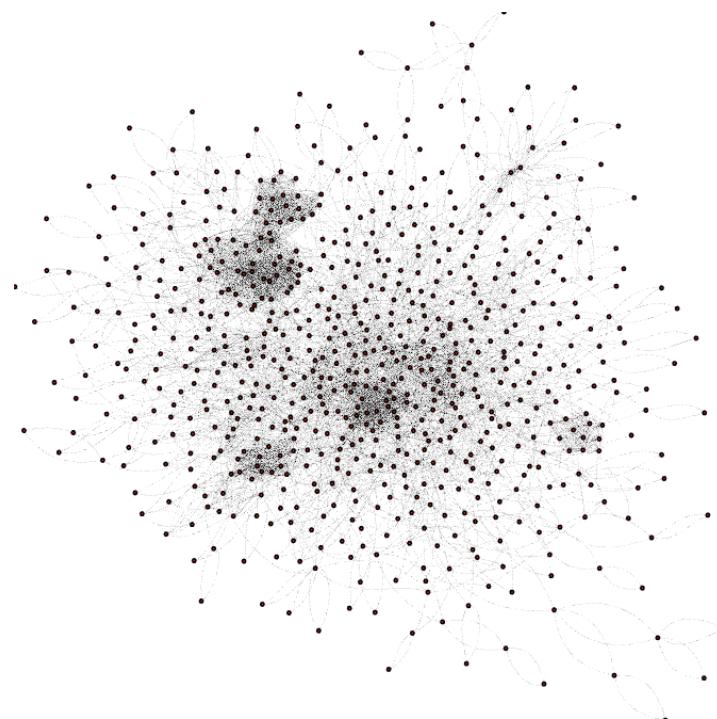


Non vediamo un incremento così spropositato del diametro o dello shortest path dato che esisteranno quelle componenti connesse che continuano a resistere per poter garantire il transito di informazioni ed inoltre non stiamo apportando modifiche a quei nodi chiave per il passaggio di informazioni (che analizziamo in seguito). Concludendo possiamo dire che la rete risulta resiliente nelle connessioni tra i TF ma non verso i geni che non potranno più essere raggiungibili tolte quelle connessioni. Questo può essere spiegato dal fatto che la rete è disassortativa e quindi non esistono altri hubs che possano fare da “serbatoio” di connessioni verso quei geni-target, quindi tolto quel nodo le connessioni cadranno. Infatti a livello biologico esistono TF specifici per specifici geni e non possono essere sostituiti con altri quindi spiegato questo comportamento.

Ora tentiamo di eliminare 50 brokers e partiamo mostrando come la Giant Component si modifica:



Si può cominciare a notare il formarsi di piccoli cluster che si distaccano dalla Giant Component e possiamo vedere chiaramente l'effetto aumentando il numero di brokers eliminati a 200:



Questo può essere spiegato dal fatto che eliminando i nodi ponte si spezzeranno quelle connessioni di quei gruppi di nodi che sfruttavano il broker per far transitare le informazioni, allora gli archi utilizzati verso quel nodo cadranno lasciando così isolato i due gruppi di nodi formando due cluster distinti.

Ora analizziamo dal punto di vista analitico quello che succede:

Giant Component Before:

Reciprocity: 0.6497509829619922

Density: 0.013278644763576361

Average Clustering Coefficient: 0.2462836843281994

Giant Component After:

Reciprocity: 0.17527747462059592

Density: 0.00015722807069665684

Average Clustering Coefficient: 0.21403011791247303

Possiamo notare come la misura di reciprocità e densità siano basse come nel caso degli hubs ma il coefficiente di Clustering non si è abbassato di tanto e questo conferma ciò che è stato osservato a livello grafico.

Si può far vedere che aumentando il numero di brokers eliminati si porterà il cluster coefficient a livelli più alti, perché eliminando i nodi ponte non starò rimuovendo la maggior parte degli archi come nel caso degli hubs ma andrò piano piano a rendere più densamente interconnessi i vicinati dei nodi togliendo archi di connessione tra i vari cluster che si stanno formando.

Infatti se andassimo ad eliminare 200 brokers addirittura andremo ad aumentare il coefficiente di clustering:

Average Clustering Coefficient: 0.26108335498519136

Analizzando ora i nodi isolati e le componenti connesse troviamo:

Nodi isolati: 4976

Nodi isolati TF: 18

Nodi isolati Target: 4958

strongly connected components 16551

In questo caso abbiamo meno nodi isolati rispetto alla rimozione degli hubs, questo perché è spiegabile dal fatto che i nodi con alta betweenness non devono avere per forza una alta degree centrality e quindi anche meno connessioni saranno rimosse. Per quanto riguarda le componenti fortemente connesse sono in numero maggiore rispetto agli hubs ma non di molto, però

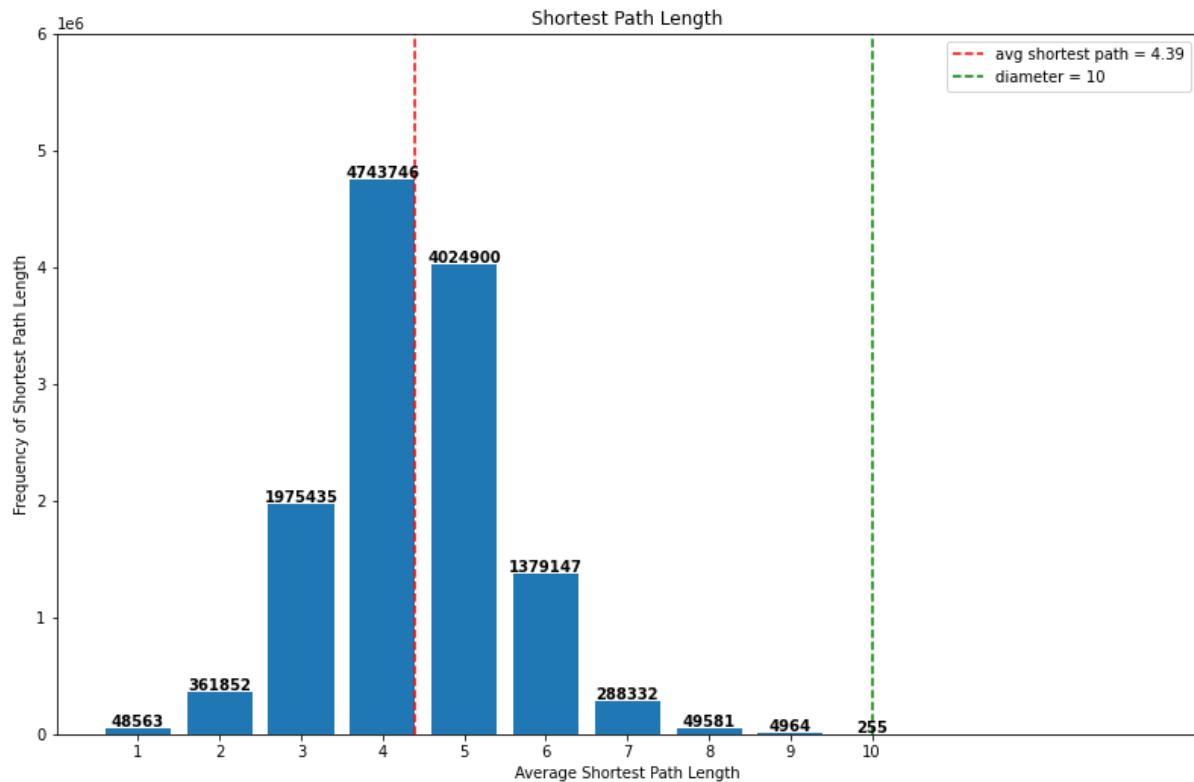
d'altro canto esse risultano maggiormente coese dato il coefficiente di clustering analizzato precedentemente.

È invece interessante vedere come siano rimasti isolati più TF rispetto a prima e questo potrebbe spiegare che diversi TF facevano affidamento su TF ponte per la loro “sopravvivenza”. Questi sono i 16 TF isolati:

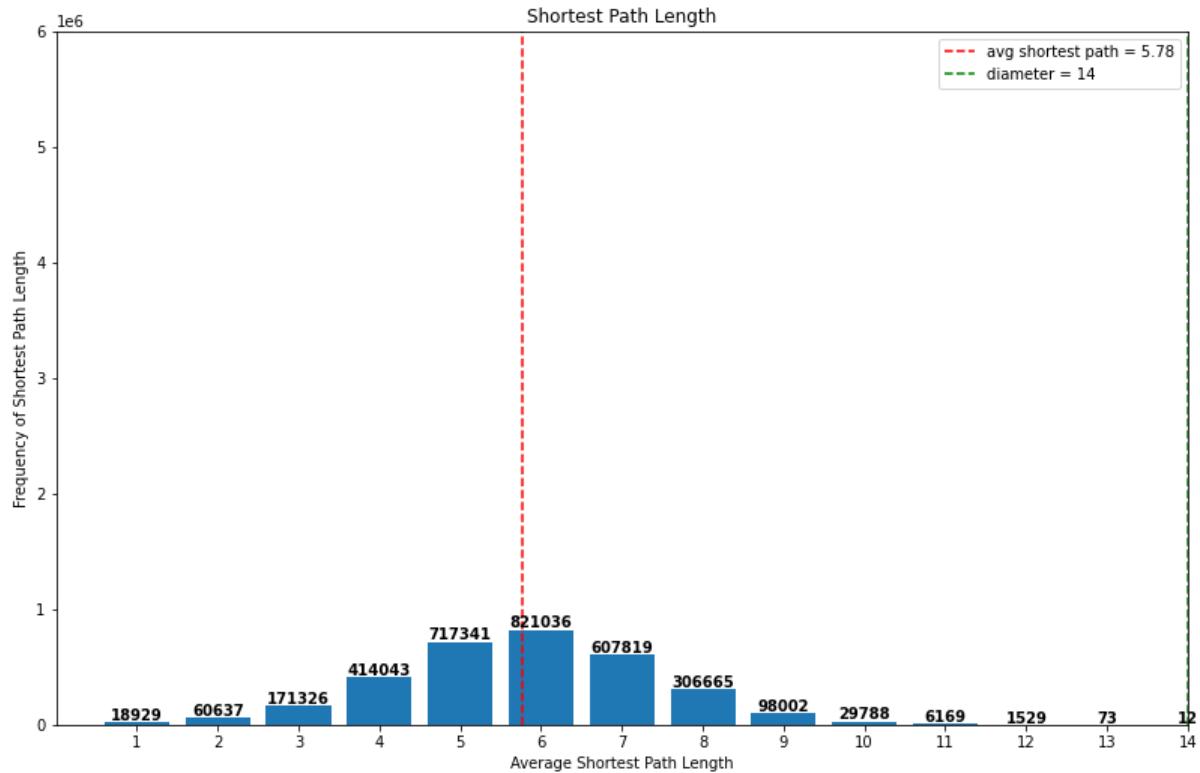
```
['Zfp523', 'Sub1', 'Sowahc', 'Ttf2', 'Tfe3', 'Mxd3', 'Zbtb7a', 'Fubp3',
'Zfml', 'Rexo4', 'Prpf4b', 'Gtf3c2', 'Foxa3', 'Zfp180', 'Brwd1',
'Ankrd7', 'Ankrd49', 'Ankrd33']
```

Vediamo di nuovo comparire *Zfp523* che utilizza il nodo ponte *Tbp* (ma anche gli altri nodi ponte *Mecom* e *Zfx*), il quale *TBP* è una proteina legante che si lega ad una particolare sequenza di DNA, il *TATA box*; allora si può intuire come *Zfp523* sfrutta *Tbp* per poter essere regolato e legare poi il DNA e se il *TBP* viene danneggiato l'intero processo smette di funzionare.

Adesso analizziamo lo shortest path e diametro:

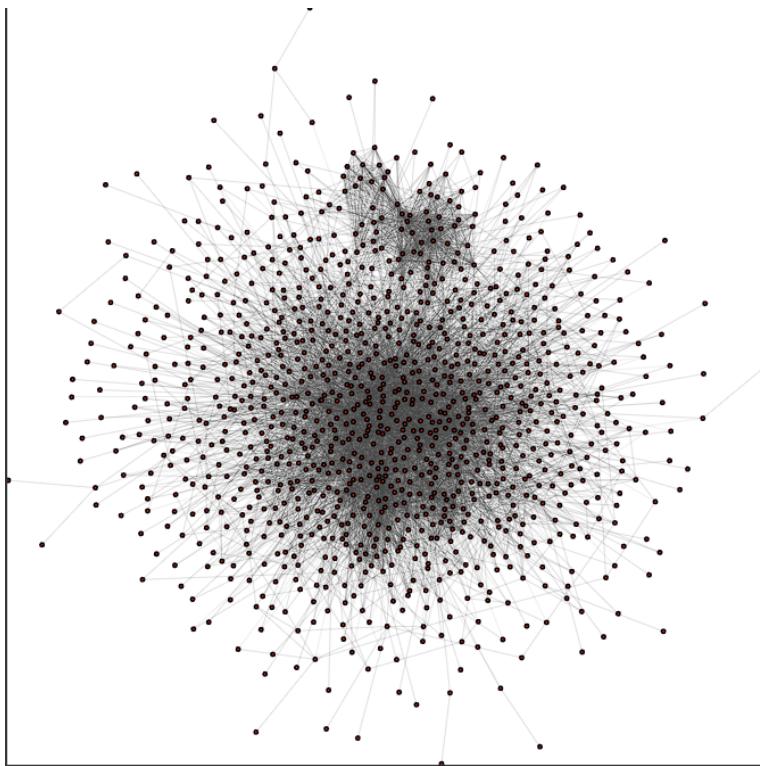


Si può notare, anche se non in modo marcato, che lo shortest-path è di poco superiore a quello calcolato prima, ma se aumentassimo i nodi brokers eliminati si può vedere che questo numero aumenti perché stiamo eliminando tutti quei shortest path possibili che portano più velocemente a destinazione. Vediamo togliendo 200 brokers come la rete si comporta:

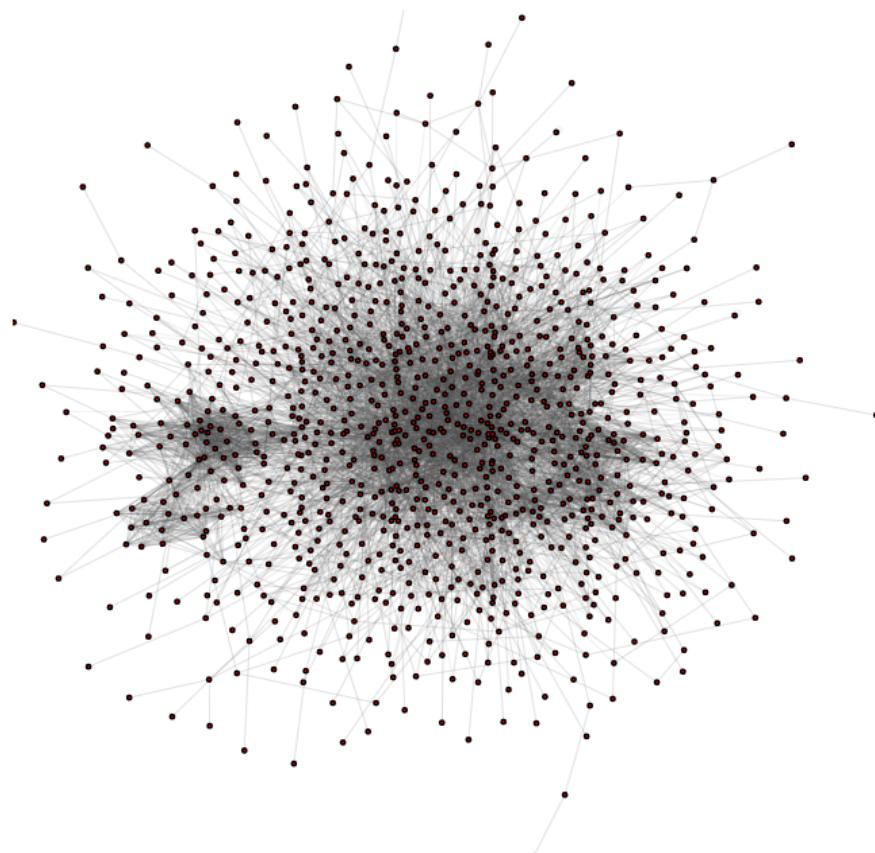


Possiamo notare come il diametro aumenti abbastanza ma comunque la shortest-path medio rimanga al di sotto dei 6 gradi di separabilità, allora si può dire che la rete risulta resiliente dal punto di vista dell'efficacia nel passaggio di informazioni, anche se il diametro risulti alto ma ciò risulta accade per poche coppie di nodi. A livello biologico questo può essere spiegato dal fatto che la rete cerca di garantire tempi brevi nella regolazione dell'espressione genica, perché viene ridondata di connessioni alternative che comunque portano alla stessa destinazione, a meno di geni isolati.

Infine proviamo ad eliminare quei nodi con alta closeness, eliminandone 50:



Analizzando la Giant Component non si notano pattern particolari come per gli spokes e gli hubs, allora proviamo ad aumentare il numero di nodi eliminati portandoli a 200:



Come si può notare anche eliminando molti nodi non stiamo modificando più di tanto la Giant Component perchè stiamo eliminando nodi con alta closeness che a loro volta non sono hubs o brokers.

Ora proviamo ad analizzare più analiticamente rispetto alle due situazioni precedenti:

```
Hubs nodes removed:  
Reciprocity: 0.251286243208155  
Density: 0.0001347295340222878  
Average Clustering Coefficient: 0.17259521132446085
```

```
Spokes nodes removed:  
Reciprocity: 0.17527747462059592  
Density: 0.00015722807069665684  
Average Clustering Coefficient: 0.21403011791247303
```

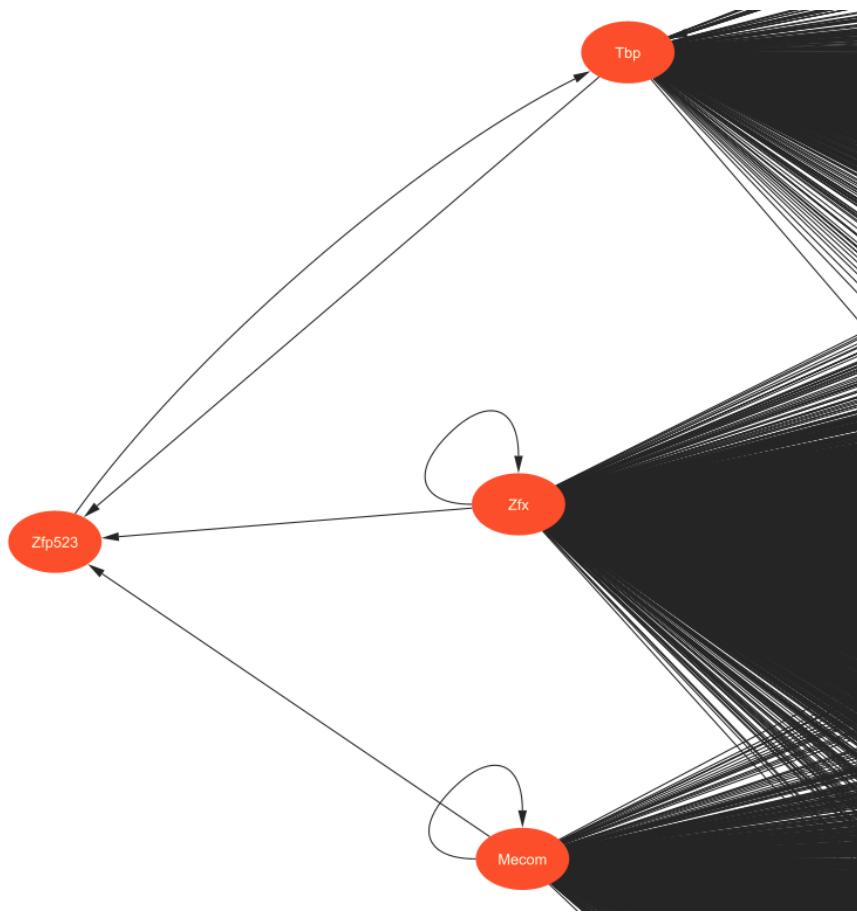
```
Closeness nodes removed:  
Reciprocity: 0.1865065478054261  
Density: 0.0001757657794039414  
Average Clustering Coefficient: 0.23039715909885106
```

Si può notare come l'indice di clustering e di densità sia più alto nel caso della closeness, perché non siamo riusciti ad eliminare completamente la giant component. Una volta tolte le connessioni che passano verso il “centro” della rete, dove questi nodi sono situati, comunque si manterranno tutti quei cammini che non passano da essi così che la rete possa continuare ad operare.

Analizzando le componenti connesse e i nodi isolati:

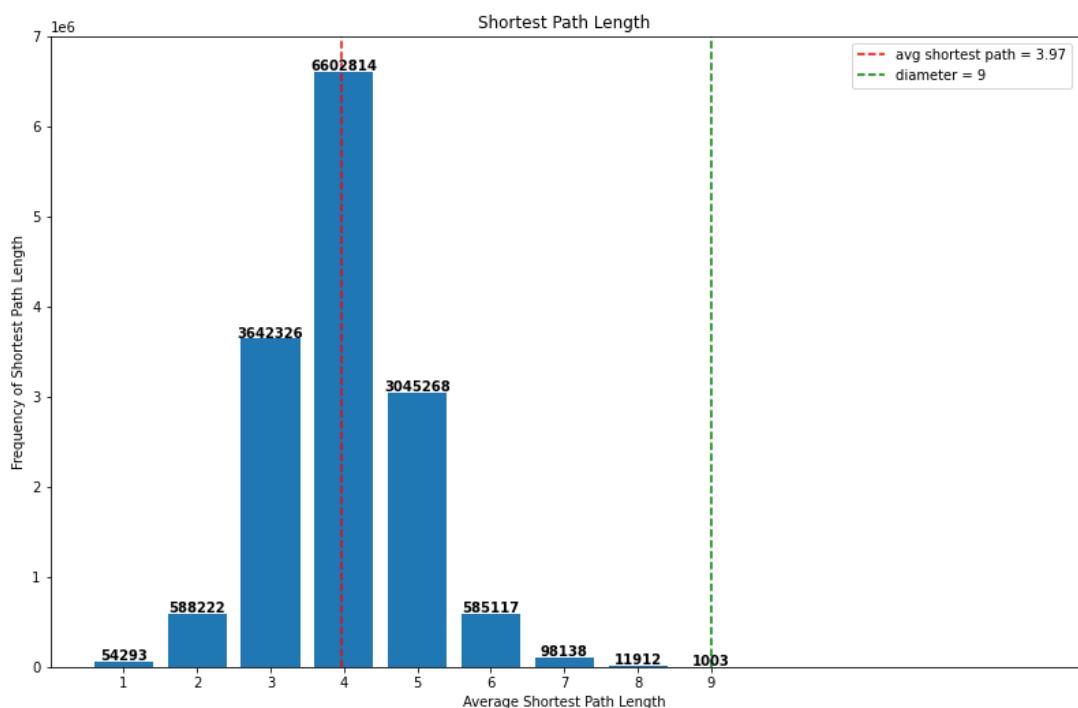
```
Nodi isolati: 4403  
Nodi isolati TF: 6  
Nodi isolati Target: 4397  
Nodi isolati TF sono ['Zfp523', 'Ttf2', 'Mxd3', 'Fubp3', 'Foxn4',  
'Brwd1']  
strongly connected components 16472
```

Avremo molti meno nodi isolati rispetto ai casi precedenti e circa lo stesso numero di componenti. Analizzando lo *Zfp523* che compare in ogni volta nelle nostre analisi possiamo notare come esso abbia una connessione diretta a *Zfx* che ha una alta closeness e questa situazione può essere visionata in questo screen:



Possiamo concludere che Zfp523 come gli altri TF isolati facciano affidamento completo ad altri TF più importanti e rimanendo senza connessioni non potrebbero più svolgere le loro funzioni.

Analizziamo infine lo Shortest-Path vediamo:



Dato il fatto che non abbiamo eliminato nodi ponte ci aspettavamo questi risultati. Si può dire che la rete risulta resiliente nel caso dovessero danneggiarsi dei TF centrali fondamentali nel trovare il più breve percorso possibile durante lo scambio di informazioni all'interno della rete e questo significa che esisteranno sufficienti nodi ponte per sopperire alla loro mancanza. A livello biologico i TF con alta closeness aiutano di certo a rendere più veloce il processo di regolazione ma non sono così importanti come possono essere l'insieme di TF ponte che bilanciano il flusso di regolazione per tutta la rete cooperando tra loro (come visto in precedenza).

# Community Detection

## Algoritmi Testati

Per il nostro dominio applicativo abbiamo dovuto escludere tra gli algoritmi di community quelli *Node-Centric* e *Group-Centric*, perché richiedono di lavorare su reti più piccole rispetto alla nostra.

Abbiamo provato ad utilizzare gli algoritmi *Network* e *Hierarchy-Centric* visti a lezione, ma solo l'approccio greedy di massimizzazione della modularità implementato dall'algoritmo di Louvain (visto in laboratorio) ha funzionato.

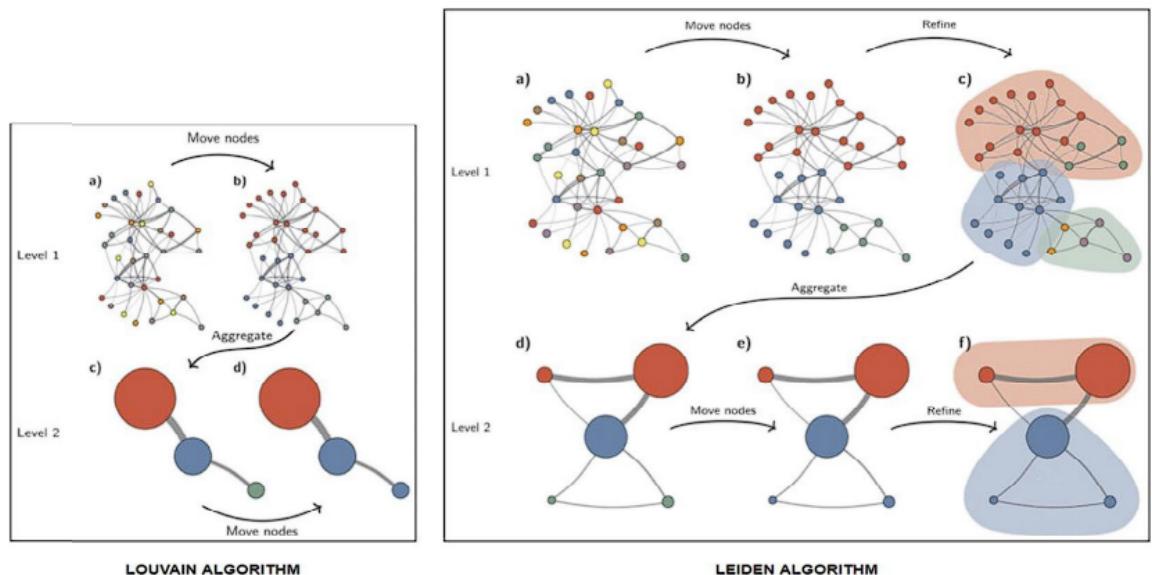
Gli algoritmi basati sullo Spectral clustering all'interno di Colab andavano in out-of-memory, mentre quello gerarchico basato sulla *edge-betweenness* rimaneva in esecuzione fin troppo tempo senza produrre alcun risultato.

## Algoritmi Cytoscape

Grazie a Cytoscape abbiamo trovato un plugin che offre degli algoritmi adatti per trattare reti biologiche, tra cui è presente anche Louvain. Andrò a descriverli brevemente:

- **Louvain:** l'algoritmo è basato sulla modularità e ha un approccio gerarchico. Opera nel seguente modo:
  - Inizialmente ogni vertice sarà assegnato verso una community a se stante
  - Ad ogni step, i vertici vengono assegnati a delle community del suo vicinato con un approccio greedy, seguendo un ordine randomico di scelta dei nodi. Ogni vertice viene spostato nella community di vicinato con la quale raggiunge il più alto contributo di modularità
  - Quando non ci sono più vertici da assegnare, le community verranno considerate come dei nuovi nodi della rete e il processo di assegnamento ripartirà da capo

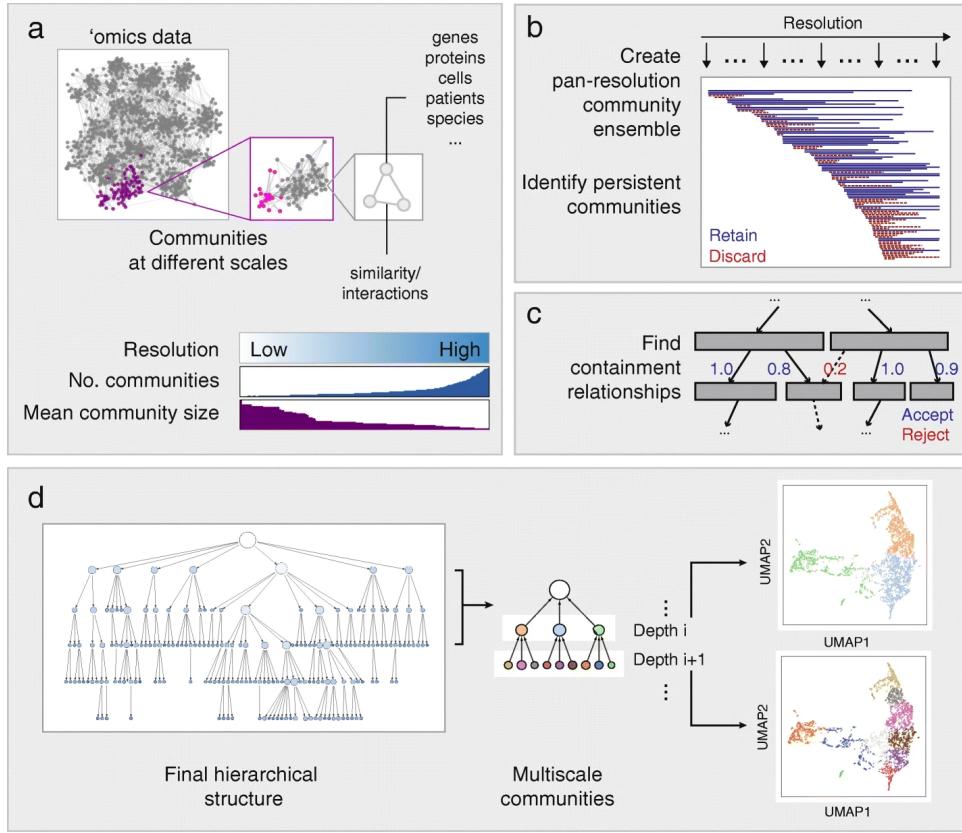
- Il tutto terminerà se c'è un solo vertice rimasto al di fuori dall'assegnamento (quando non si è riusciti a massimizzare la modularità) o quando la modularità non può essere incrementata ulteriormente al prossimo step
- Leiden:** è simile a Louvain, ma cerca di migliorare alcuni suoi limiti. Introduce l'idea di poter effettuare uno split delle community dopo la fase di local moving (assegnamento di un nodo ad una community) andando ad eseguire quest'ultima all'interno delle nuove community trovate e cercando di trovare delle sotto-community così da separarle dalla community originale.  
Con questo passaggio si trovano delle community meglio connesse. Inoltre ad ogni step considera non tutti i nodi per l'assegnamento ma solo quelli considerati instabili (che hanno il potenziale di cambiare community) impiegando meno tempo nell'esecuzione.  
Infine, per rendere Leiden computazionalmente più efficiente, al posto di selezionare la community di assegnamento con il miglior guadagno di modularità, si sceglierà in modo random tra quelle nel vicinato del nodo: questo permette un buon rapporto tra qualità delle community trovate e performance, perché scegliendo casualmente sarà comunque possibile trovare una buona community (esploro più possibilità) e scegliere un hub che porti a convergenza l'algoritmo.



- Hidef:** è un algoritmo che fa da *wrapper* per gli algoritmi di Leiden e Louvain. Lavora basandosi sul concetto dell'*omologia persistente* che permette di identificare strutture robuste nei dati, calcolando le caratteristiche topologiche di uno spazio a diverse risoluzioni spaziali.

Caratteristiche più persistenti vengono rilevate su un'ampia gamma di scale spaziali e si ritiene che rappresentino con maggiore probabilità le vere caratteristiche dello spazio sottostante piuttosto che artefatti di campionamento, rumore o particolare scelta di parametri. L'algoritmo opera in tre fasi:

- Per iniziare il grafo in ingresso viene formulato secondo una rete di similarità computata sulla somiglianza dei loro profili biologici (geni, proteine, cellule...) e si cercherà di trovare gruppi densamente connessi dal punto di vista biologico, *massimizzando il valore di modularità tramite uno dei due algoritmi citati a priori*.
- Nella seconda fase sfruttando la Jaccard similarity si trovano community simili che si ripetono facendo variare il parametro di *pan resolution*, che è il coefficiente moltiplicativo della modularità e permette di avere differenti dimensioni delle community con più o meno nodi all'interno di essi. Quelle community più persistenti saranno portate alla fase finale dell'algoritmo.
- Nell'ultima fase le community persistenti sono analizzate per identificare le relazioni di contenimento: se la quantità di nodi condivisi tra due community supera una determinata soglia fissata allora esse verranno considerate come due community distinte ma in overlap tra loro. Si avrà una gerarchia di community innestate e in overlap dove avremo alla radice la community che conterrà tutti i nodi della rete ed ogni arco dell'albero indicherà che quella community è contenuta nella community discendente.
- Ecco un riassunto visivo dei passaggi appena illustrati:



- **Infomap:** un algoritmo di community detection ottimale con una funzione obiettivo che minimizza la quantità di informazione richiesta per esprimere il movimento di un *Random Walker* all'interno di un grafo, sfruttando la codifica di Huffman per la compressione.

La codifica di Huffman usa un metodo specifico per scegliere la rappresentazione di ciascun simbolo, risultando in un codice senza prefissi (cioè in cui nessuna stringa binaria di nessun simbolo è il prefisso della stringa binaria di nessun altro simbolo) che esprime il carattere più frequente nella maniera più breve possibile.

Un Random Walker è un processo stocastico che permette di esplorare in modo casuale il nostro grafo utilizzando la probabilità di transitare verso un altro nodo tramite un matrice di transizione di una catena di Markov.

Tramite questo passaggio abbiamo codificato la nostra rete assegnando ad ogni nodo un parola chiave data dalla matrice di transizione, ma i nodi di una rete reale sono troppi quindi avremo il problema di trovare un modo per ridurre i codici associati ad essi. Partendo dall'assunzione che in una rete reale esistono delle regioni in cui una volta entrati sarà poco probabile transitare fuori da essa facendo diventare il movimento tra regioni raro, allora si potranno usare i codici Huffman: utilizzare un prefisso per ognuna di queste regioni da porre davanti ai codici di ogni

nodo, in modo tale da poterli riutilizzare per altri regioni (che avranno prefissi diversi) e diminuire il numero di codici diversi per nodo. L'obiettivo quindi sarà trovare il miglior partizionamento del grafo in termini di community per minimizzare il numero di codici richiesti per nodo, quindi di trovare l'ottimo della nostra funzione obiettivo. Trovare pochi cluster ci farebbe tornare al problema di partenza, ma averne troppi farebbe esplodere la lunghezza del prefisso, quindi i cluster dovranno essere della dimensione ottimale per ottenere un numero di community che minimizzi la funzione obiettivo.

- **Oslom:** è un algoritmo multi-purpose in grado di trattare grafi orientati, archi pesati, overlap delle community, gerarchie e dinamica nelle community. È basato sull'ottimizzazione di una funzione di fitness espressa tramite la significatività statistica dei cluster rispetto alle fluttuazioni casuali (cambiamenti repentini di valori nel tempo inevitabili in un processo random).

La *significatività statistica* è la probabilità di trovare un cluster in un grafo random che non presenta alcuna struttura per la formazione di community; questo grafo è chiamato **null model** ed è lo stesso utilizzato nella definizione di modularità adottato da Newman and Girvan. Gli step che l'algoritmo eseguirà saranno:

- Prima di tutto si cerca di trovare dei cluster significativi impostando un parametro di tolleranza (**p-value che controlla il numero di cluster**) da superare per determinare se quel cluster è inaspettatamente poco probabile da trovare in null model (con la stessa distribuzione di grado) e se supera tale tolleranza di inattesa probabilità, allora significa che sarà un cluster significativo.

Da questo processo possono originarsi cluster che sono in overlap tra loro, dato che ogni cluster viene costruito indipendentemente dagli altri: ogni cluster è creato considerando mano a mano tutti i vertici del grafo che sono compatibili con la significatività statistica attesa per quel cluster nel null model. Un vertice viene definito *compatibile* se condivide molti più archi con i vertici all'interno del cluster che si sta costruendo, rispetto a quello che ci si aspetta di trovare nel null model, così da considerare la relazione tra quel vertice e quel cluster inaspettatamente forte.

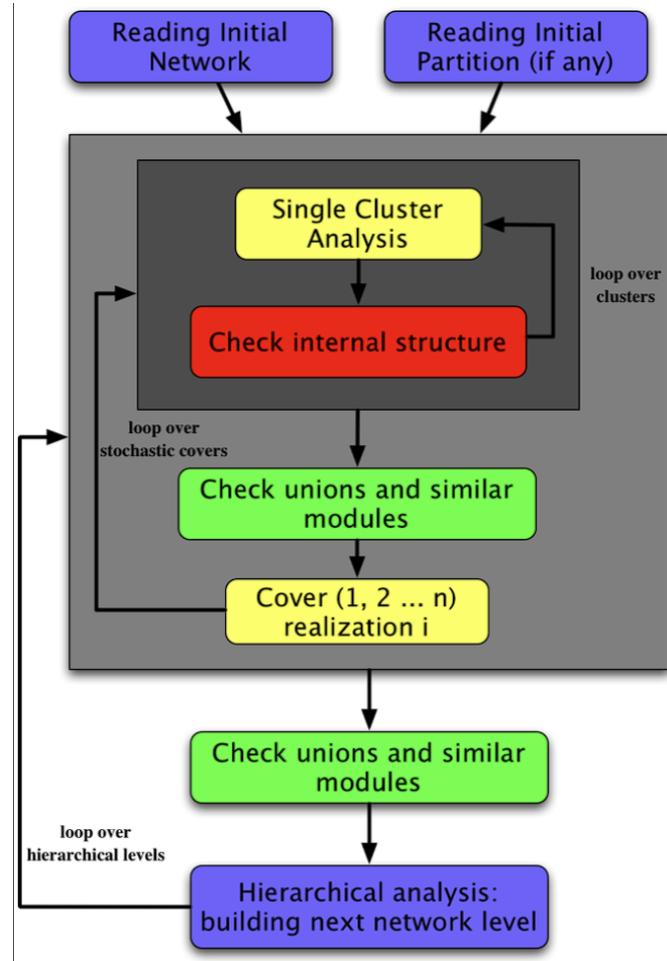
- Secondariamente si analizza l'insieme di cluster risultante cercando di trovare dei sotto-cluster significativi **minimali** (**nessuna significatività di struttura interna in termini di cluster**) all'interno di ciascuno di essi: tramite un soglia del livello di significatività (**coverage parameter controlla la dimensione dei cluster**): se l'unione dei due sotto-cluster trovati supera questa soglia rinuncerò ai nuovi sotto-cluster altrimenti li considerò entrambi dei cluster significativi minimali.
- Dopo aver trovato questi cluster minimali dobbiamo capire se, presi a coppie, l'unione di essi produca ancora cluster minimali (con il metodo usato prima), altrimenti si mantiene il cluster minimale più grande (questo permette anche di eliminare cluster simili). Infine essendo una copertura potrà essere riemessa come input nella fase iniziale dell'algoritmo per ottenere cover più fini e migliorare la stocasticità dell'algoritmo.
- Infine c'è la fase in cui viene identificata una struttura gerarchica dei cluster, escludendo a priori i nodi homeless, che non appartengono a nessun cluster.

Innanzitutto viene creata una nuova rete avente cluster minimali come nodi, con archi presenti tra quei nodi rappresentativi di un cluster che sono collegati tra di loro nel grafo originario, pesati dal numero di nodi tra i 2 cluster.

Per quegli archi nel grafo originario che hanno un estremo in un cluster ed uno nell'altro allora il loro contributo sarà:  $1/(V_x * V_j)$ .

Una volta che la rete è stata creata si ritorna a cercare altri cluster significativi minimali della gerarchia fino a che non si producono più cluster, lavorando sulla rete appena creata.

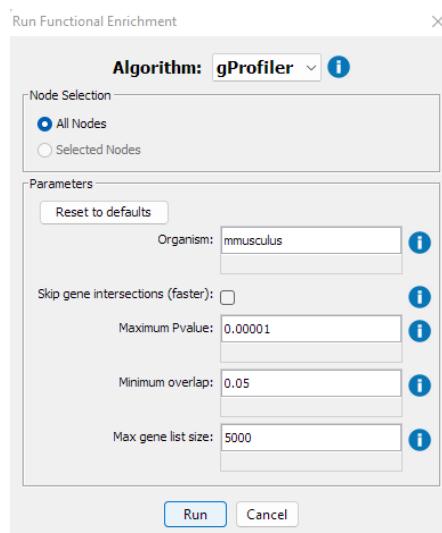
Ecco uno schema riassuntivo del suo funzionamento:



## Algoritmo Ottimale

L'algoritmo scelto è l' OSLOM sia perché fornisce i migliori risultati nelle analisi comparative, mostrate in seguito, sia perché è l'unico a permettere di eseguire in simultanea un'analisi gerarchica e di overlap delle community, che sarà utile per capire il livello di importanza dei vari gruppi funzionali (gerarchia) e quali nodi svolgono funzioni diverse dato che sono presenti in più community (overlapping).

Per poter effettuare i dovuti confronti abbiamo sfruttato una funzione offerta dallo stesso plugin di Cytoscape, da cui abbiamo usufruito di questi algoritmi, per poter effettuare un “arricchimento funzionale” cosicché ogni community venisse confrontata con una base di conoscenza che conteneva tutti i gruppi funzionali (un gruppo di due o più geni che svolgono funzioni simili all'interno della cellula) fino ad ora catalogati per l'organismo del *Mus Musculus*. Questa è la schermata di cytoscape:



Questa base di conoscenza può essere vista come una sorta di parziale *ground truth* perché diverse community non sono state riconosciute e presupponiamo non esistano ancora abbastanza dati riguardanti l'organismo in analisi.

Prima di effettuare i dovuti confronti sarebbe giusto descrivere il significato dei vari parametri di confronto:

- *Recognized\_%*: la percentuale di community riconosciute tramite arricchimento funzionale rispetto al totale di community trovate
- *Unique\_communities*: community distinte riconosciute tramite arricchimento funzionale
- *Recognized\_community*: community totali riconosciute tramite arricchimento funzionale

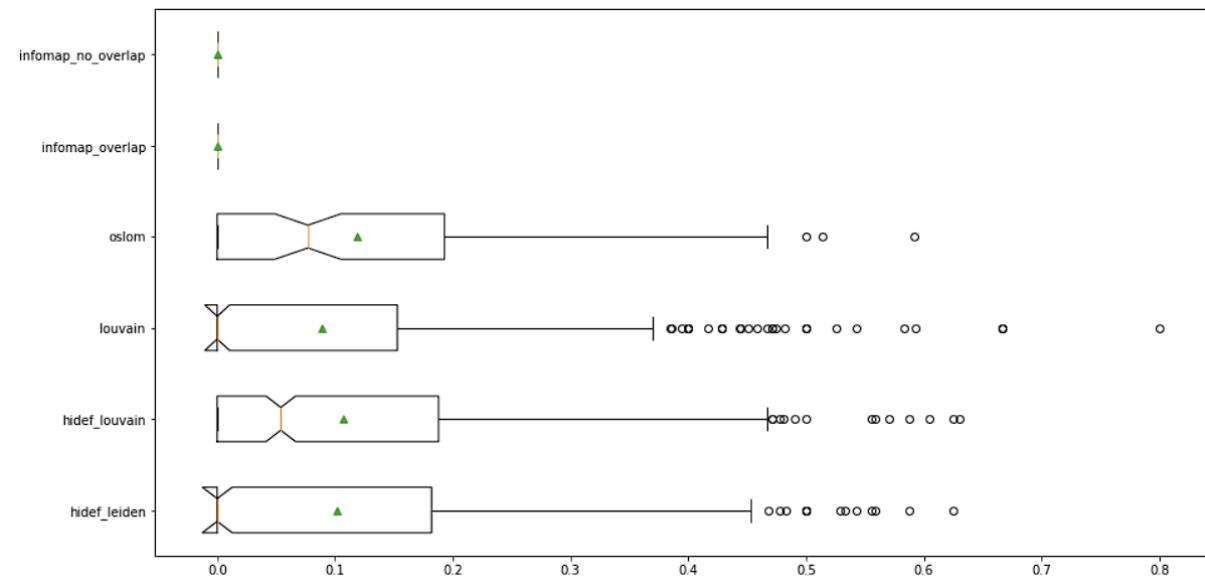
- *Not\_Recognized\_community*: community totali non riconosciute tramite arricchimento funzionale
- *Total\_communities*: community totali che sono state trovate dall'algoritmo di community
- *Unique\_%*: *Unique\_communities / Total\_communities*
- *Granularity*: numero delle più piccole community riconosciute, tramite arricchimento funzionale, tra 1 e 10 nodi
- *Duplicates*: numero di duplicati trovati tramite arricchimento funzionale, ossia stessi gruppi funzionali riconosciuti su community diverse
- *Avg\_liability*: misura media di tutti i valori di **overlap** (*verosimiglianza*) di ogni community (si considerano le community totali ed anche quelle non riconosciute nel calcolo della media). Essa viene misurata guardando la quantità di overlap dei nodi presenti in una community e il gruppo funzionale (*ground truth*), presente nella base di conoscenza, che abbia il massimo numero possibile di etichette, nei nodi, corrispondenti a quelle che possiamo trovare nella community analizzata. Può essere visto come un indice di performance dell'algoritmo di community nel trovare le community più coerenti possibili in termini di funzionalità che svolgono all'interno dell'organismo.
- *Avg\_liability\_std*: rappresenta la deviazione standard della media del calcolo precedente
- *Recognized\_Avg\_liability*: misura media della quantità di overlap considerando esclusivamente quelle riconosciute e non quelle totali.

Presentiamo ora queste analisi effettuate utilizzando i parametri di default di ogni algoritmo e impostando il seed a 1 per rendere confrontabile ogni risultato dato che si ha per alcuni metodi una componente stocastica:

algorithm	recognized %	unique communities	recognized communities	not recognized communities	total communities	unique %	granularity	duplicates	avg likelihood	avg likelihood	stdev	recognized avg likelihood
oslom	0.591304	66	68	47	115	0.970588	3	2	0.119165	0.140805	0.201629	
hdef_louvain	0.508108	204	282	273	555	0.723404	28	57	0.107526	0.136694	0.211621	
hdef_leiden	0.456349	170	230	274	504	0.739130	25	48	0.102206	0.139686	0.223965	
louvain	0.388781	151	201	316	517	0.751244	46	44	0.088785	0.143651	0.228368	
infomap_overlap	0.000000	0	0	142	142	0.000000	0	0	0.000000	0.000000	0.000000	
infomap_no_overlap	0.000000	0	0	11322	11322	0.000000	0	0	0.000000	0.000000	0.000000	

L'algoritmo ottimale tra quelli presentati risulta l'OSLOM perché presenta rispetto agli altri algoritmi vantaggi in quasi tutti i fronti tranne che per il numero di community totali, il avg likelihood e il granularity. D'altro canto siamo più interessati ad ottenere community che non abbiano un alto numero di duplicati e una likelihood maggiore, rispetto a trovare più community ma

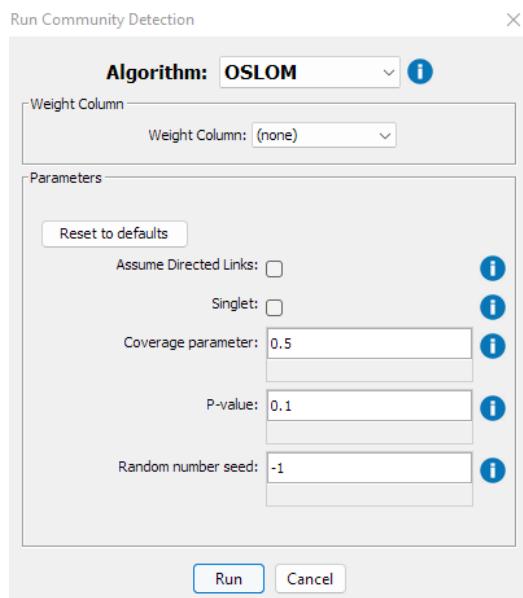
meno coerenti con la nostra ground truth. Si può vedere che si avrà una recognized avg likelihood più bassa delle altre perchè ci saranno molte meno community riconosciute rispetto agli altri casi, ma dall'altro canto gli altri algoritmi hanno molte più community non riconosciute che fanno crollare così la misura di avg. Per quanto riguarda la variazione dei risultati di overlap avremo situazioni simili per tutti gli algoritmi. Si può intuire anche visualmente questa situazione facendo un box-plot dell'overlap:



Avremo che l'OSLOM ha meno outliers rispetto agli altri algoritmi e anche media e mediana sono verso valori più alti come ci si poteva aspettare. Si può notare, anche che nella tabella, che la varianza è elevata e questo fa intendere che comunque sia difficile per qualsiasi algoritmo trovare una partizione di community ottimale data la complessità del grafo.

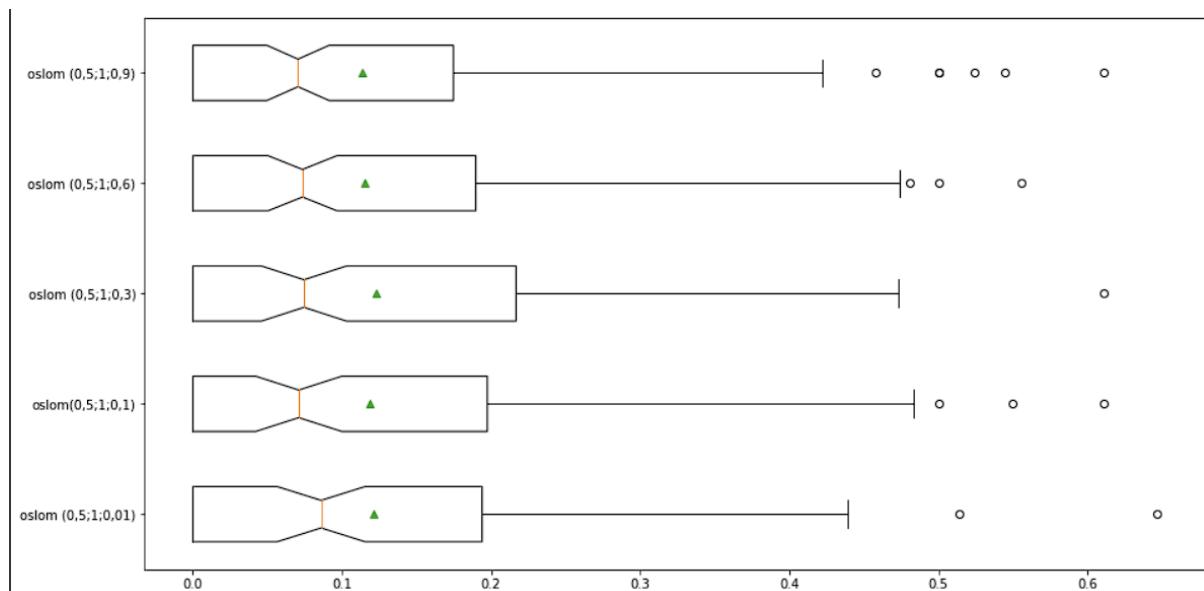
## Tuning Parametri OSLOM

Ora andremo ad effettuare il tuning dei parametri del p-value e del coverage parameter (mantenendo sempre il seed fissato):



Innanzitutto partiamo tenendo fissato il parametro Coverage Parameter e facendo variare il P-value :

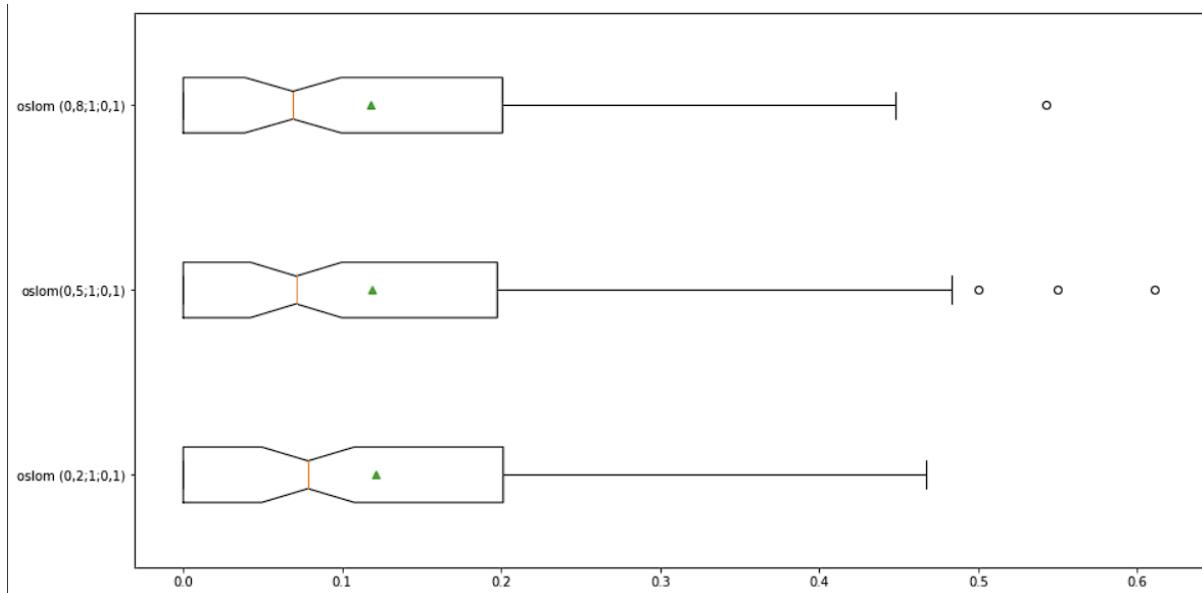
algorithm	recognized %	unique communities	recognized communities	not recognized communities	total communities	unique %	granularity	duplicates	avg likelihood	avg likelihood stdev	recognized avg likelihood
oslom (0,5;1,0,01)	0.635514	64	68	39	107	0.941176	2	3	0.121393	0.134486	0.191015
oslom (0,5;1,0,9)	0.586207	97	102	72	174	0.950980	5	5	0.113908	0.139304	0.194314
oslom (0,5;1,0,3)	0.584507	81	83	59	142	0.975904	2	2	0.122768	0.143152	0.210036
oslom (0,5;1,0,6)	0.572289	85	95	71	166	0.894737	7	9	0.114952	0.139738	0.200863
oslom(0,5;1,0,1)	0.568966	64	66	50	116	0.969697	2	2	0.118983	0.144065	0.209121



Il valore ottimale del parametro di P-value risulta il **valore 0.3** perchè ha una più alta likelihood e una più alta percentuale di community distinte. Come già specificato, si può vedere che il p-value farà aumentare il numero di cluster.

Passiamo ora all'analisi il coverage parameter tenendo fisso invece il P-value:

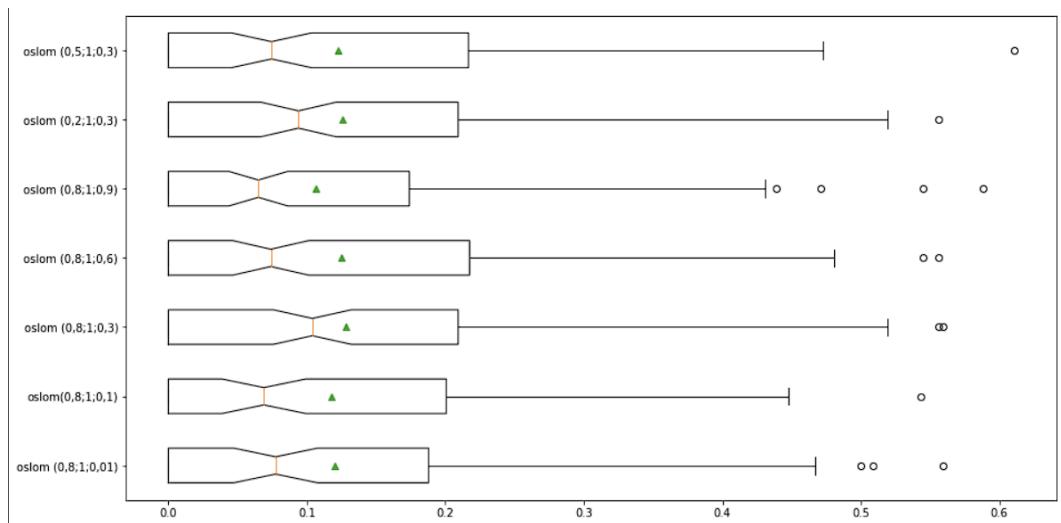
algorithm	recognized %	unique communities	recognized communities	not recognized communities	total communities	unique %	granularity	duplicates	avg likelihood	avg likelihood stdev	recognized avg likelihood
oslom (0,2;1,0,1)	0.618644	69	73	45	118	0.945205	1	3	0.120661	0.133837	0.195041
oslom (0,8;1,0,1)	0.592593	63	64	44	108	0.984375	1	1	0.117685	0.137857	0.198594
oslom (0,5;1,0,1)	0.568966	64	66	50	116	0.969697	2	2	0.118983	0.144065	0.209121



Il valore ottimale del parametro del coverage parameter risulta il **valore 0.2**, con la più alta percentuale di community distinte, riconosciute e più alta likelihood. Purtroppo avremo più duplicati ma, come vedremo in seguito, figurano sempre nello stesso gruppo funzionale.

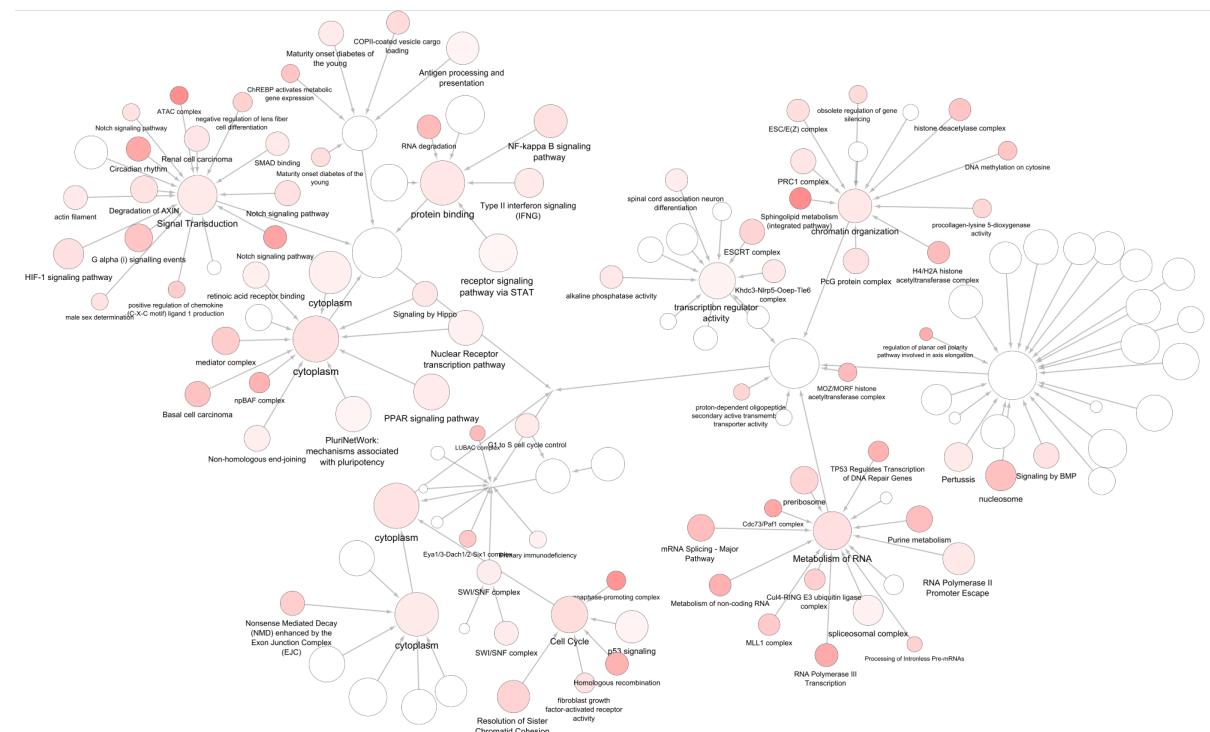
Proviamo ora a trovare il giusto compromesso tra i due parametri facendoli variare insieme:

algorithm	recognized %	unique communities	recognized communities	not recognized communities	total communities	unique %	granularity	duplicates	avg likelihood	avg likelihood stdev	recognized avg likelihood
oslom (0,8;1,0,3)	0.625000	78	85	51	136	0.917647	3	4	0.128338	0.140312	0.205341
oslom (0,2;1,0,3)	0.616438	84	90	56	146	0.933333	4	5	0.125815	0.138382	0.204100
oslom (0,8;1,0,1)	0.604167	55	58	38	96	0.948276	0	3	0.120396	0.143903	0.199276
oslom (0,8;1,0,1)	0.592593	63	64	44	108	0.984375	1	1	0.117685	0.137857	0.198594
oslom (0,5;1,0,3)	0.584507	81	83	59	142	0.975904	2	2	0.122768	0.143152	0.210036
oslom (0,8;1,0,9)	0.563636	86	93	72	165	0.924731	7	6	0.106655	0.132317	0.189226
oslom (0,8;1,0,6)	0.548387	83	85	70	155	0.976471	4	2	0.124845	0.146190	0.227659



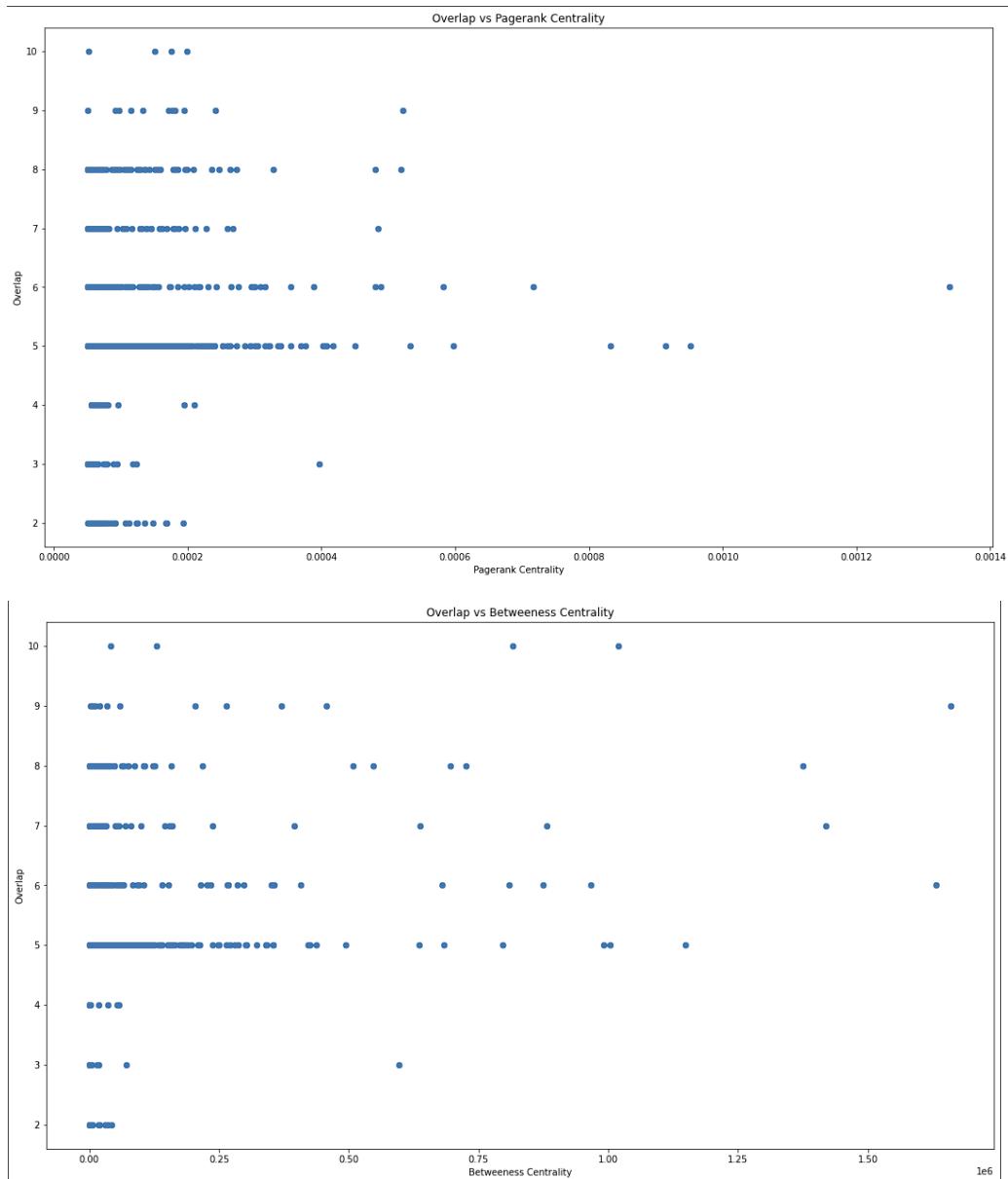
Infine abbiamo scelto come trade-off **p-value=0.3** e **coverage parameter=0.8** perchè permettono di ottenere una più alta percentuale di community riconosciute, oltre ad una migliore likelihood, anche se con una percentuale di unique più bassa, ma analizzando l'elenco dei duplicati scopriamo che a ricomparire è sempre il citoplasma, quindi in pratica il numero di duplicati torna pari ad 1.

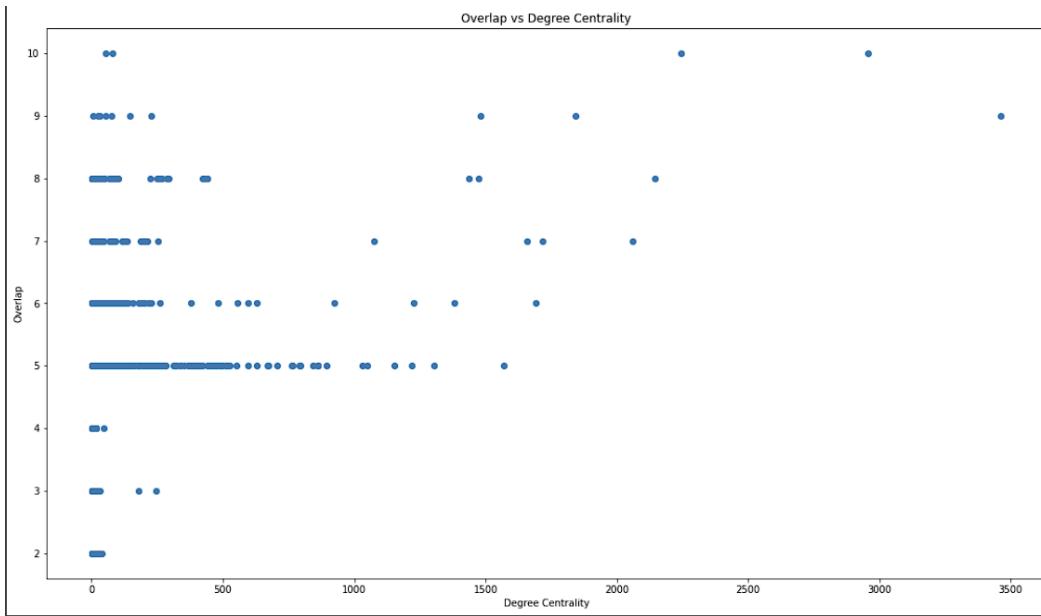
Di seguito la rappresentazione gerarchica ottenuta dall'algoritmo:



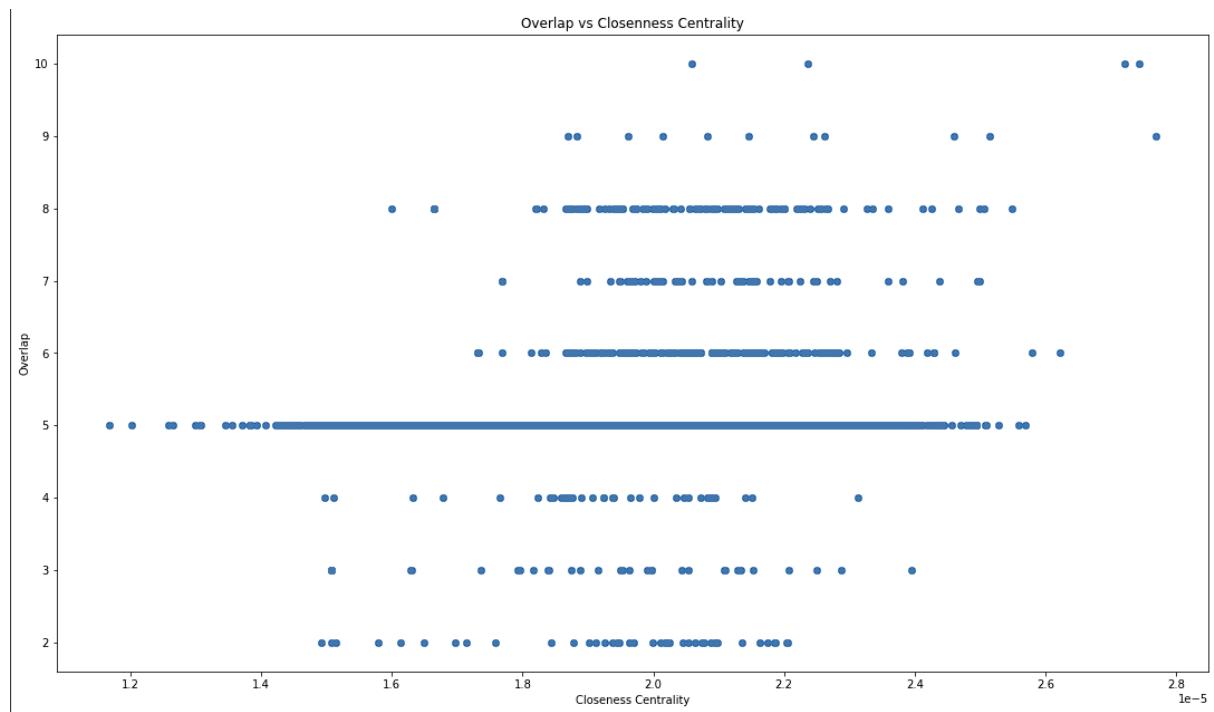
## Overlap nodi

Sarà interessante analizzare quali nodi hanno overlap per trovare quelli che svolgono più di una funzione all'interno della cellula e che quindi sono contenuti in più *cluster genici* (un gruppo di due o più geni che codificano per la stessa proteina o per proteine simili tra di loro). Innanzitutto possiamo confrontare l'overlap tra le diverse misure di centralità:





Dal confronto con queste centralità non abbiamo riscontrato alcun pattern specifico quindi possiamo affermare che esse siano scorrelate dal fenomeno dell'overlap ed invece il caso della closeness risulta molto interessante da analizzare:



Si può notare come ci sia un pronunciato spostamento dei plot sulla parte destra del piano nel caso di overlap maggiore di 5, separati da un “segmento” di plot lungo l’asse orizzontale dove l’overlap è pari a 5 ed al di sotto di esso tutti i plot saranno posizionati sull’area sinistra del piano. Questo

comportamento può essere spiegato intuitivamente dal fatto che i nodi con alta closeness sono quei nodi da cui passa la maggior parte dell'informazione di tutta la rete e quindi saranno anche quelli che saranno preposti a svolgere la maggior parte delle funzioni all'interno della cellula, dato che dovranno comunicare con più gruppi funzionali possibili. Poi avremo quei nodi con overlap pari a 5 che chiamerei come nodi di *intermezzo* i quali non hanno un ruolo preciso di importanza all'interno della cellula, quindi possono variare di molto la loro efficienza nel diffondere informazioni a tutta la rete. Infine quelli con overlap più basso saranno anche quei nodi con meno funzioni e quelli che hanno un ridotto scambio di informazioni limitato ai pochi gruppi funzionali di cui fanno parte.

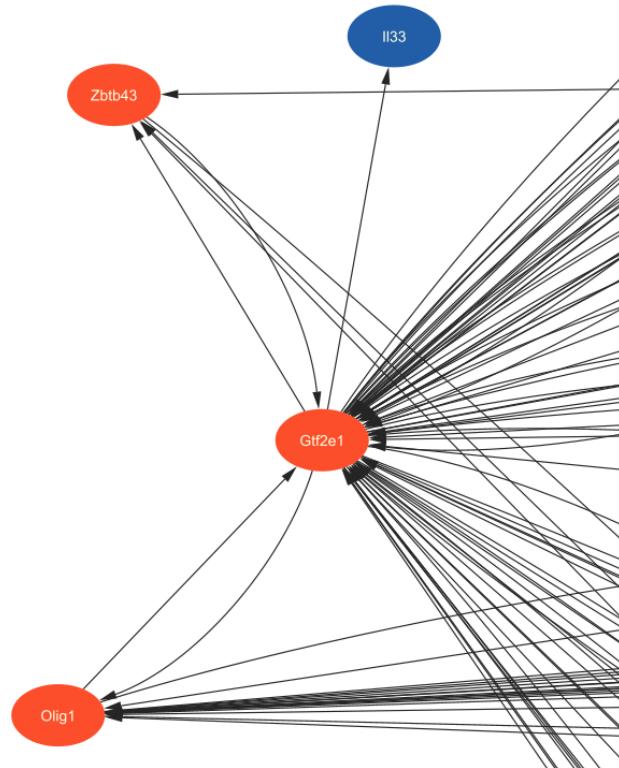
Adesso analizziamo quei nodi che hanno un più alto grado di overlap cercando di capire a livello biologico come si comportano andando però ad eliminare quelle community che non sono state riconosciute, così da rendere l'analisi più consistente. Questi due nodi sono quelli con l'overlap più alto:

Gene	Overlap	Gene_Type
Gtf2e1	6	TF
Mre11a	4	Target

Partiamo ad analizzare il nodo *TF Gtf2e1* e i gruppi funzionali di cui fa parte:

Cytoplasm  
Metabolism of RNA  
RNA Polymerase II Promoter Escape  
Khdc3-Nlrp5-Ooep-Tle6 complex  
transcription regulator activity  
PluriNetWork: mechanisms associated with pluripotency

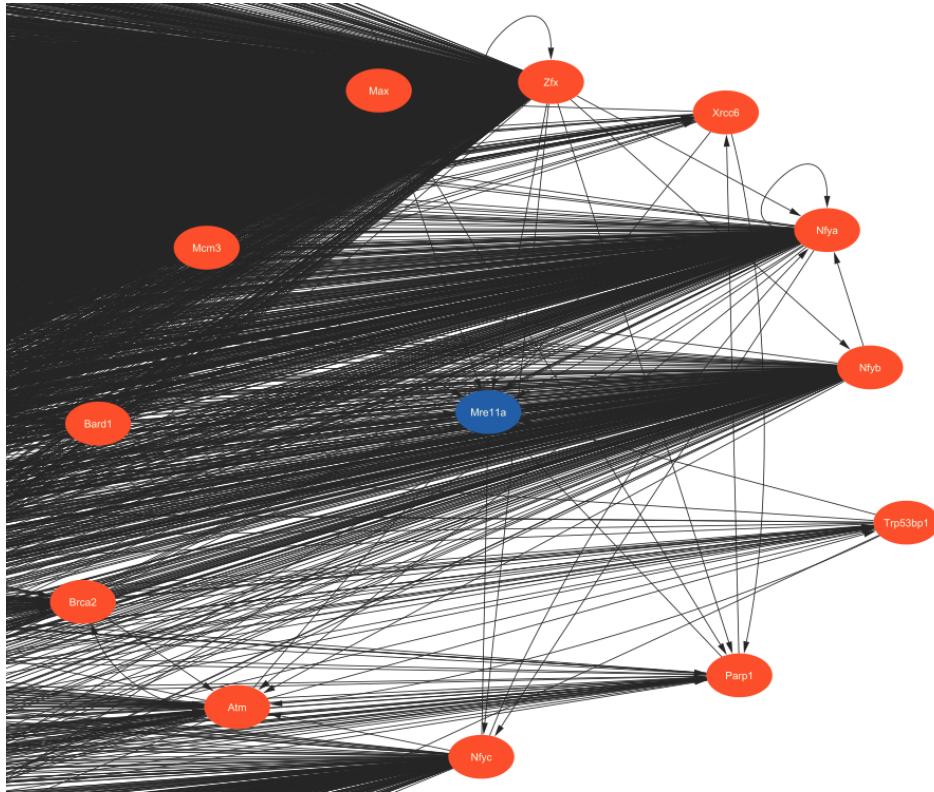
Infatti il *Gtf2e1* sta per **General transcription factor IIE subunit 1** e i general transcription factor sono una classe di fattori di trascrizione proteica che fanno parte del complesso multiproteico che avvia la trascrizione (insieme alla RNA polimerasi e il mediatore). I GTF sono anche intimamente coinvolti nel processo di regolazione genica, nel metabolismo RNA (formazione e degradazione), nella regolazione del RNA polimerasi (ce ne più di uno) e meccanismo associato alla pluripotenza (descrive l'abilità di produrre risposte diverse a stimoli esterni, adattandosi alla situazione corrente in modo da conferire la capacità ad una cellula di differenziarsi in tipi cellulari differenti). Si può far vedere come questo TF sia a sua volta regolato da tanti altri TF anche se esso è l'avviatore del processo di trascrizione:



Invece analizzando il nodo target-gene *Mre11a* vediamo che esso appartiene:

Homologous recombination  
 Cell Cycle  
 Non-homologous end-joining  
 Cytoplasm

Il gene *Mre11a* codifica per proteine nucleari coinvolte nel meccanismo di riparazione del DNA, nel ciclo cellulare (è la serie di eventi che avvengono in una cellula eucariote tra una divisione cellulare e quella successiva.). Mostriamo tutti i TF connessi ad esso:



Possiamo vedere direttamente l'interazione tra TF che si regolano tra di loro o autoregolano, prima di andare a regolare direttamente la trascrizione di quel gene specifico.

## Estrazione Community

Dalla gran quantità di community ottenute, è nata la necessità di operare delle scelte per estrarre quelle che assumono un comportamento più rilevante, sulla base di diverse metriche, in maniera da poterle analizzare nel dettaglio.

In generale, per ciascun termine di paragone, verranno considerate, tra le community più significative, quelle che siano state riconosciute dall'esecuzione dell'*arricchimento funzionale* di Cytoscape, in maniera tale da poter effettuare delle analisi e considerazioni anche sulla base della verosimiglianza ottenuta.

Di seguito sono elencate le metriche utilizzate, ed evidenziate le community da estrarre per essere analizzate nel dettaglio, anche in base a quante diverse metriche spiccano per rilevanza.

- **Likelihood**

Di seguito sono elencati le 5 community con la maggiore confidenza da parte del *functional enrichment* di rappresentare il gruppo funzionale indicato.

Di queste saranno estratte *Sphingolipid metabolism*, *ATAC complex* e *Notch signaling pathway*, che ritroveremo successivamente anche in altre metriche

Community	Size	Likelihood
<b>Sphingolipid metabolism (integrated pathway)</b>	28	0.559
<b>ATAC complex</b>	14	0.556
anaphase-promoting complex	17	0.519
<b>Notch signaling pathway</b>	38	0.448
Cdc73/Paf1 complex	12	0.429

- **Size**

Tra quelle riconosciute, la community più grande conta più di 3.000 nodi, e si tratta del *citoplasma*. La rete conteneva anche altre 3 community riconosciute come citoplasma, ed è stata scelta quella che, oltre ad essere più grande, ha anche un valore di likelihood maggiore.

Sono inoltre state selezionate *protein binding*, *Signal Transduction* e *Cell Cycle*, che ritroveremo ancora in seguito

Community	Size	Likelihood
<b>cytoplasm</b>	3.461	0.145
<b>protein binding</b>	2.517	0.117
<b>Signal Transduction</b>	950	0.104
<b>Metabolism of RNA</b>	754	0.156
receptor signaling pathway via STAT	730	0.05
<b>Cell Cycle</b>	545	0.166
transcription regulator activity	505	0.064

### • Assortativity

Come visto in precedenza la rete è disassortativa, e allo stesso modo lo è la maggior parte delle community:

Assortative: 8.088 %

Disassortative: 88.24 %

Neutre: 3.68 %

Average: -0.3522

Std dev: 0.2799

Tra le community da analizzare sono state scelte sia quelle con valori di assortatività molto alti, tra cui *proton-dependent oligopeptide secondary active transmembrane transporter activity* che risulta completamente disassortativa. Sono inoltre state estratte anche le 2 community più assortative della rete, che rappresentano un caso molto raro, con valori di rispettivamente 0.66 di *Sphingolipid metabolism*, e 0.37 di *Notch Signaling Pathway*, che era anche tra le community con maggior likelihood.

Tra le community più disassortative compare anche *RNA Polymerase II Promoter Escape*, responsabile della regolazione della RNA polimerasi, che ci siamo ritrovati più volte nelle precedenti analisi. Essa contiene infatti i geni del tipo *Polr2\_* e *Gtf\_*.

Community	Assortativity	Likelihood
proton-dependent oligopeptide secondary active transmembrane transporter activity	-1	0.2
Nuclear Receptor transcription pathway	-0.80	0.069
RNA Polymerase II Promoter Escape	-0.78	0.115
Maturity onset diabetes of the young	-0.77	0.162
Pertussis	-0.76	0.104
Sphingolipid metabolism (integrated pathway)	+0.66	0.559
Notch signaling pathway	+0.37	0.448

- **Distribuzione dei feedback-loop**

Data l'importanza dei feedback loop, o regulatory circuits, all'interno della rete, è importante analizzare come essi si distribuiscono tra le community. *Cytoplasm* prevale sulle altre community anche in questo caso, ma è interessante notare come manchino alcune delle community di size più alta. Su queste community ci si aspetta di trovare porzioni di rete molto fitte che svolgono un ruolo centrale in quelle che sono le funzioni svolte dal corrispondente gruppo funzionale.

Community	# feedback loop
cytoplasm	1194
Metabolism of RNA	1084
transcription regulator activity	416
Nuclear Receptor transcription pathway	388

<b>Signal Transduction</b>	355
Chromatin organization	351
<b>RNA Polymerase II Promoter Escape</b>	306

- **Distribuzione degli Hub**

Analizzando la distribuzione all'interno delle community dei nodi con almeno 1000 di degree, si ritrovano le 2 community più grandi, *cytoplasm* e *protein binding*, e *PluriNetWork*, che ritroveremo in seguito tra le community con più TF di self-regulation.

Community	# hubs
<b>cytoplasm</b>	12
<b>protein binding</b>	3
Nuclear Receptor transcription pathway	2
<b>PluriNetWork: mechanisms associated with pluripotency</b>	2

- **Distribuzione dei TF di self-regulation**

È stato anche importante analizzare la distribuzione dei TF di self-regulation, avendo visto l'importanza che essi ricoprono all'interno della rete. Qui ritroviamo community già viste nelle precedenti metriche, e sono per lo più le community più grandi ad averne in maggioranza, oltre a *PluriNetWork* che presentava già molti hub.

Community	# self TF
<b>Cytoplasm</b>	43
<b>protein binding</b>	18
<b>Signal Transduction</b>	15
<b>PluriNetWork: mechanisms associated with pluripotency</b>	6

- **Distribuzione degli Housekeeping TF**

Allo stesso modo è stata analizzata infine la distribuzione degli housekeeping TF, e anche qui ricompaiono le community più grandi, ma è *protein binding* ad avere la meglio su *cytoplasm* in questo caso.

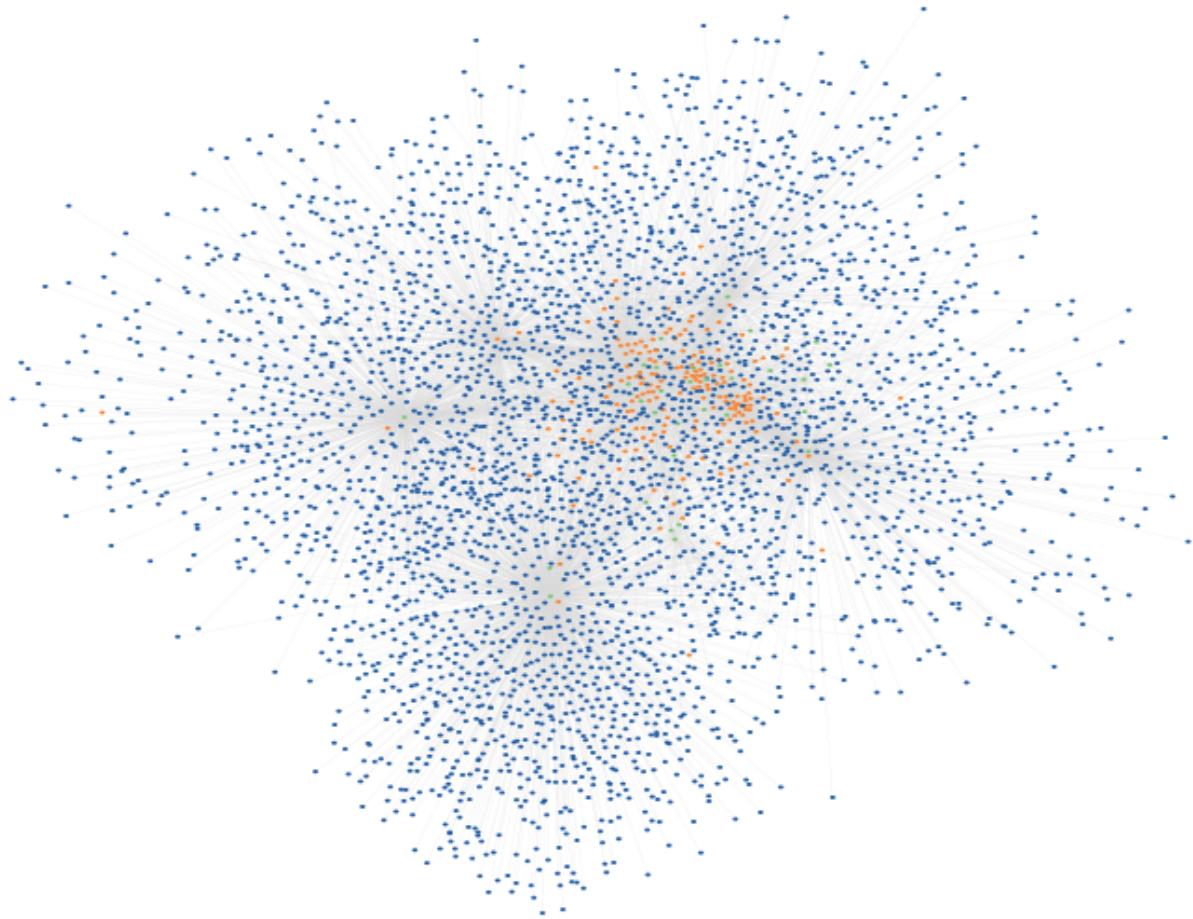
Community	# housekeeping
<b>protein binding</b>	7
<b>cytoplasm</b>	3
Resolution of Sister Chromatid Cohesion	2
<b>Cell Cycle</b>	2

## Analisi Community

Di seguito saranno esposte le community scelte tramite il processo precedente, analizzate nel dettaglio, esponendo le funzionalità di ciascuna e le particolarità riscontrate per ognuna di esse.

	Name	Size	Assortativity	Likelihood	Highest Degree	Highest Betweenness
	cytoplasm	3461	-0.504071	0.145	Ctcf, Esrrb, Zfx, Pparg, Klf4	Pou5f1, Ctcf, Esrrb, Tfcp2l1, Pparg
	protein binding	2517	-0.178216	0.117	Jun, Arnt, Ahr, Stat1, Stat5a	Rela, Stat1, Jun, Ahr, Stat5a
	Signal Transduction	950	-0.384523	0.104	Hnf1a, Creb1, Ep300, Crebbp, Hif1a	Hnf1a, Ctnnb1, Ep300, Crebbp, Creb1
	Metabolism of RNA	754	-0.192363	0.156	Tbp, Polr2e, Polr2h, Polr2f, Foxp3	Foxp3, Tbp, Ercc3, Polr1c, Polr2e
	Cell Cycle	545	-0.356972	0.166	Trp53, Kntc1, Mcm3, Mcm2, Mcm7	Trp53, Mcm4, Kntc1, Brca2, Rb1
proton-dependent oligopeptide secondary active...		10	-1.000000	0.200	Cdx2, Eomes, Hlrf, Hbegf, Furin	Cdx2, Cldn3, Furin, Sis, Hbegf
Sphingolipid metabolism (integrated pathway)		28	0.667329	0.559	Lass2, Lass3, Lass4, Lass5, Sgpp2	Lass2, Ppap2c, Sgpp2, Sgpp1, 9130409l23Rik
Notch signaling pathway		38	-0.174873	0.448	Notch2, Notch3, Notch4, Dtx1, Hey1	Notch2, Notch4, Notch3, Dtx1, Lfng
ATAC complex		14	-0.417905	0.556	Kat2b, Yeats2, Dr1, Zzz3, Kat2a	Kat2b, Mbip, Ivl, Kat2a, Abcb1b
Nuclear Receptor transcription pathway		389	-0.802110	0.069	Esrrb, Pparg, Nr0b2, Nr1i3, Nr1h3	Esrrb, Pparg, Nr4a2, Nr1h3, Nr1i3
PluriNetWork: mechanisms associated with pluri...		433	-0.463445	0.058	Tfcp2l1, Pou5f1, Esrrb, Nanog, Rnf2	Pou5f1, Tfcp2l1, Esrrb, Zfp42, Sall4

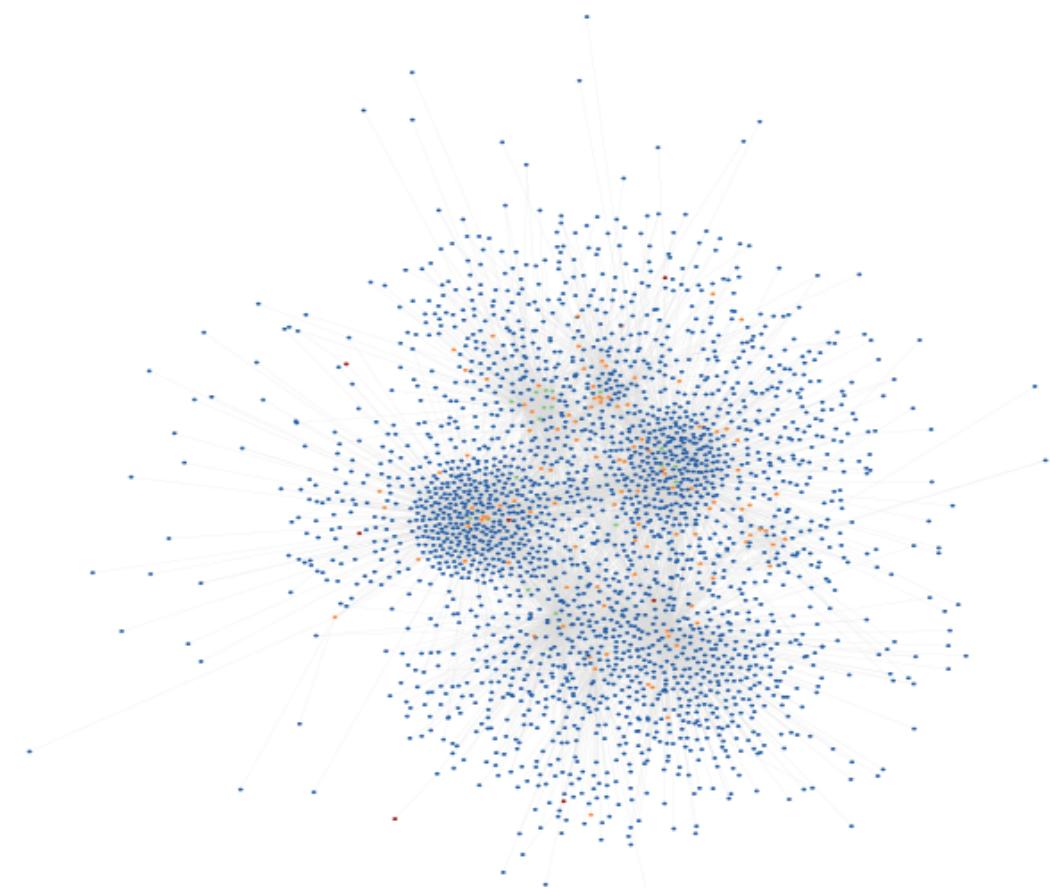
- **Cytoplasm**



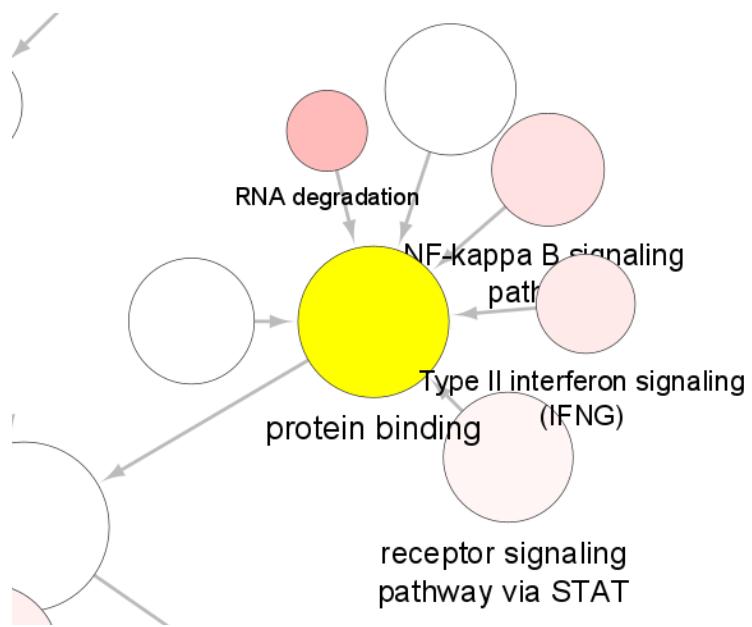
Nel citoplasma si svolgono le principali attività della vita cellulare (funzioni legate al metabolismo, processi di sintesi...), e la sua importanza è confermata dal fatto che rappresenta la più grande community della nostra rete, oltre ad essere quella con più hub e self-regulation TF.

Come si nota dall'immagine, la rete presenta una fitta distribuzione di TF al centro, ed è costituita a sua volta da altri 4 cluster alle estremità: tutti di principalmente target genes, con un TF al centro, che abbiamo visto essere proprio i TF con edge-betweenness più alta, che nella community in questione fanno proprio da broker.

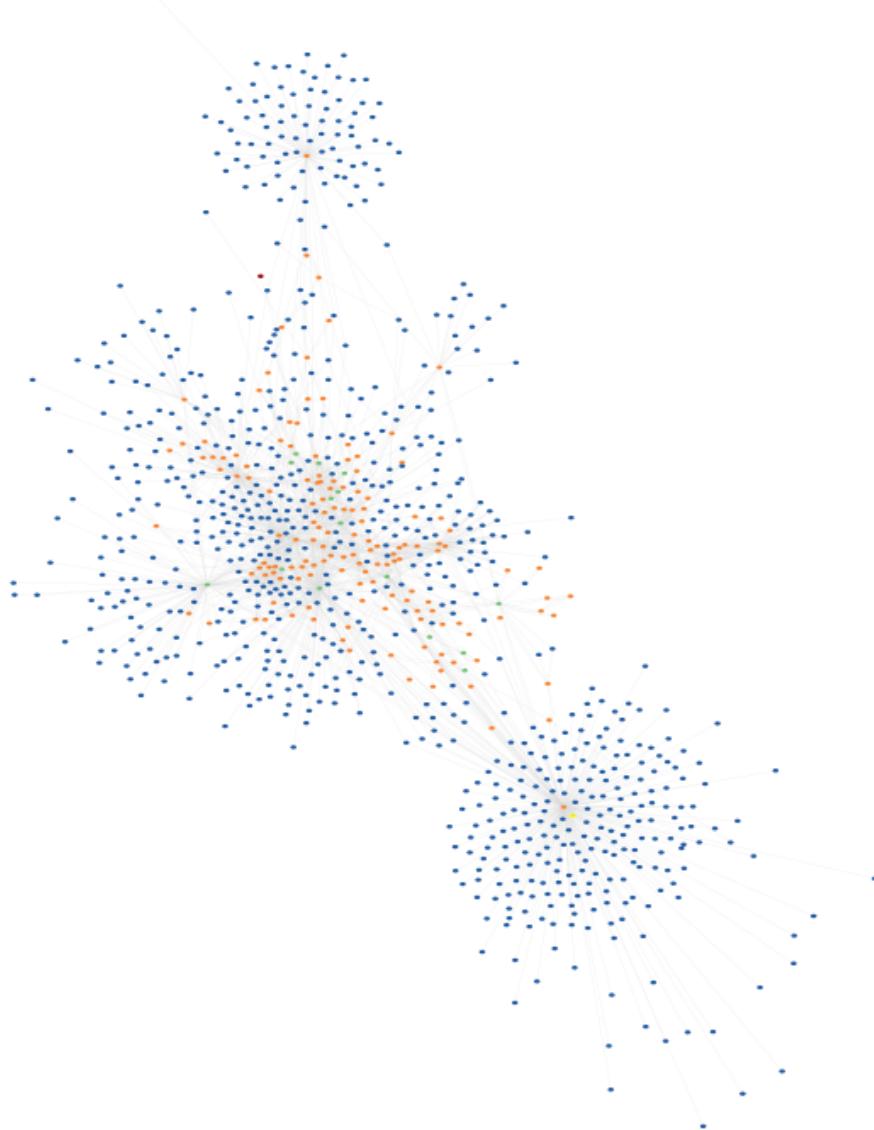
- Protein Binding



Responsabile dei legami con il DNA (iniziazione della trascrizione, riparazione del DNA...), si tratta della seconda community più grande e quella con più housekeeping genes, oltre a presentare un gran numero di hub e self-regulation TF.



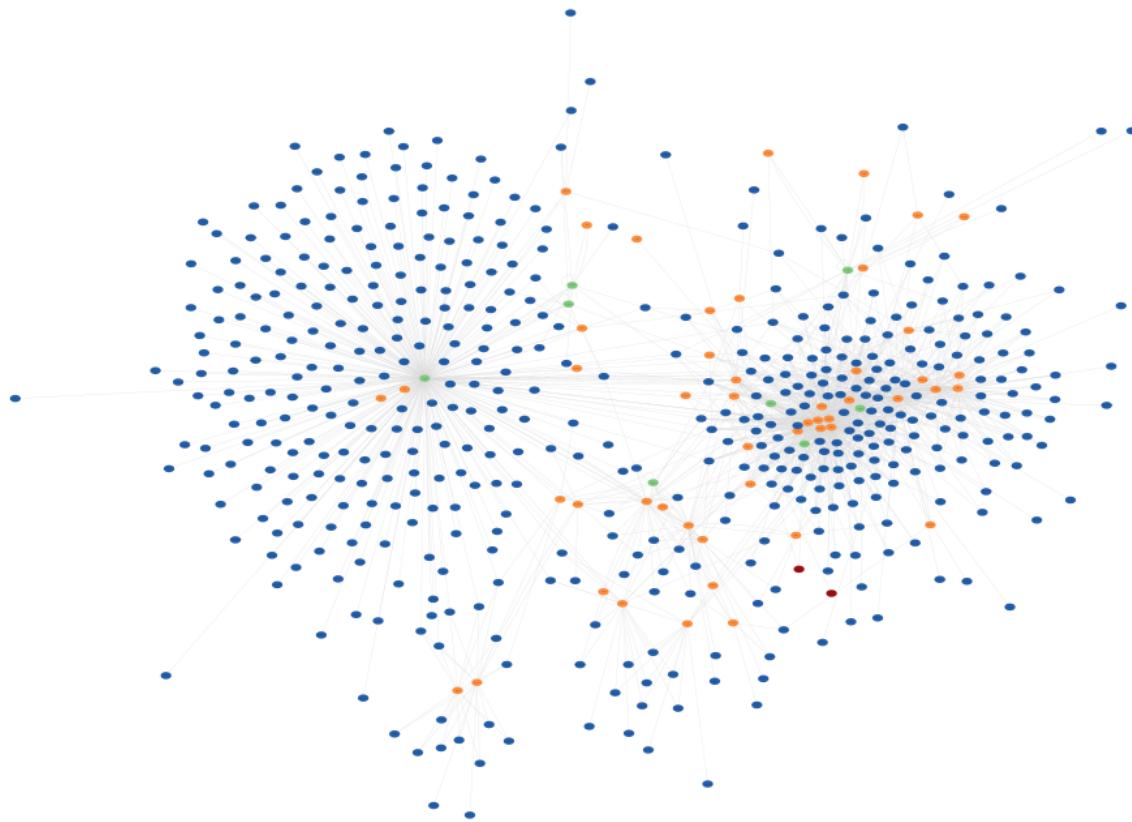
- Signal Transduction



Processo in cui una molecola segnale attiva un recettore cellulare determinando l'attivazione di una via biochimica all'interno della cellula che genererà una risposta, è tra le community più grandi, con più self-regulation TF e feedback loop, tutte proprietà che le assicurano robustezza e resistenza. Comprende diversi cluster, tra cui quello per il *Ritmo Circadiano*, *Notch Signaling Pathway* e *ATAC Complex*, nei quali c'è tendenzialmente un TF centrale che lega agli altri.

Il ruolo più centrale è svolto da *Hnf1a*, il gene con più alta Degree e Betweenness centrality, che domina il subcluster a sud della rete, connettendosi anche a molti TF degli altri cluster.

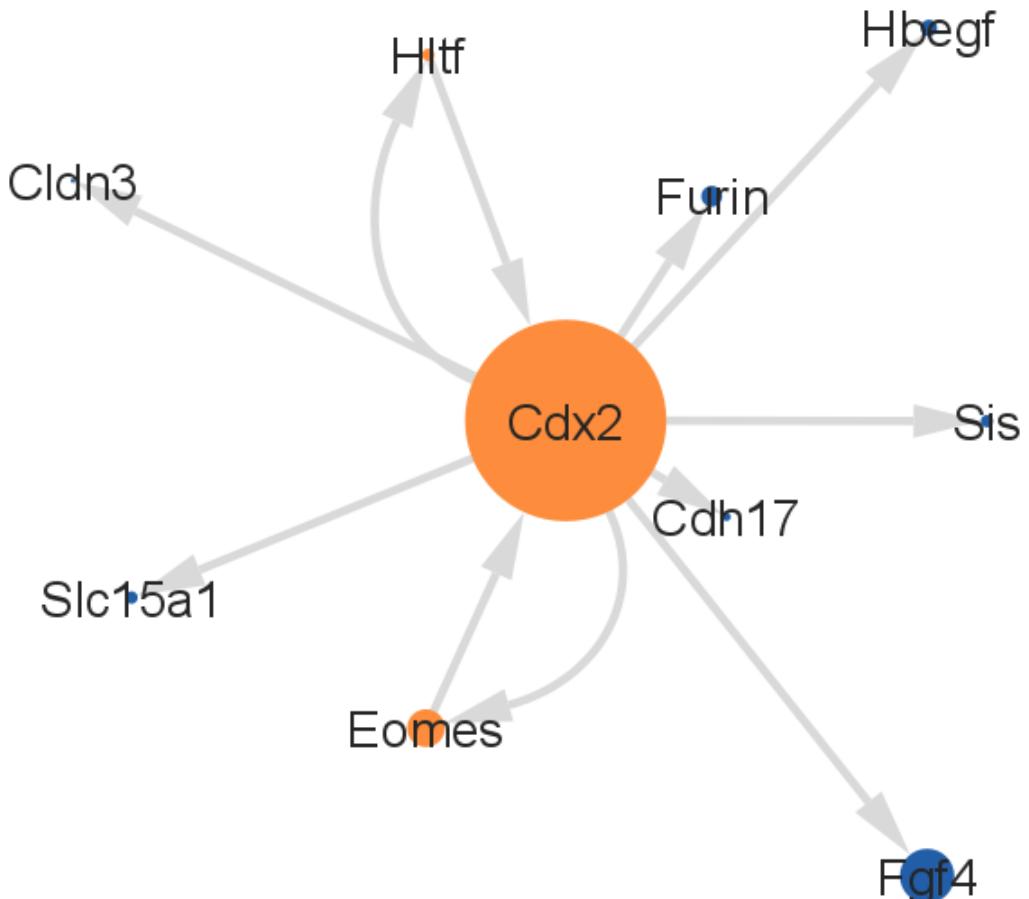
- **Cell Cycle**



Responsabile del processo in 4 stage della cellula, in cui aumenta di dimensione, copia il proprio DNA, prepara la divisione cellulare e la esegue per mezzo della mitosi.

È tra le community più grandi e presenta molti housekeeping genes e regulatory circuits, a testimonianza del fatto che l'attività svolta è di fondamentale importanza per la riproduzione cellulare e di conseguenza per l'organismo.

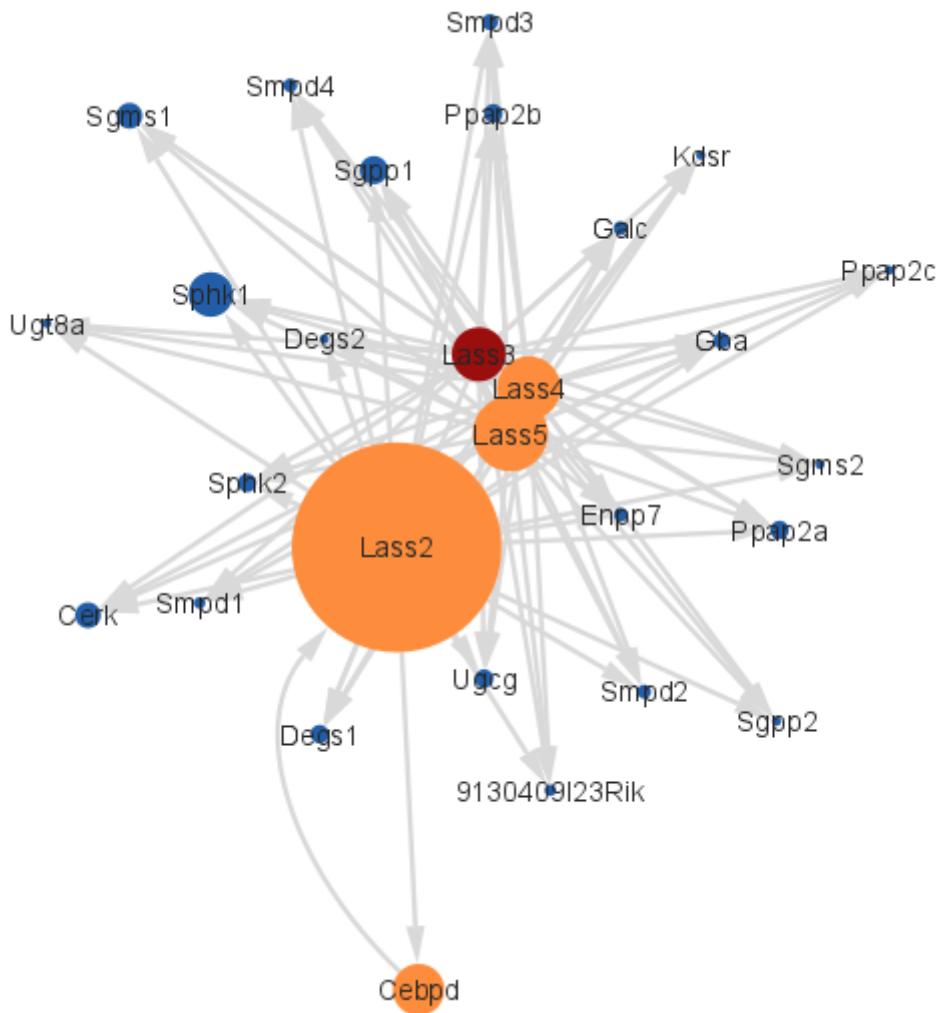
- Proton-dependent oligopeptide secondary active transmembrane transporter activity



Il trasporto di polipeptidi (proteine) è un fenomeno biologico effettuato da specifici trasportatori che si trovano in numerosi e differenti organismi, tra cui batteri ed umani.

Si tratta dell'unica community totalmente disassortativa della rete, con  $-1$ . Tutti i nodi esterni comunicano con l'unico gene centrale, *Cdx2*. Si tratta per la maggior parte di target gene, ma troviamo anche due TF, *Hltf* e *Eomes*, che generano altrettanti feedback loop con il nodo centrale.

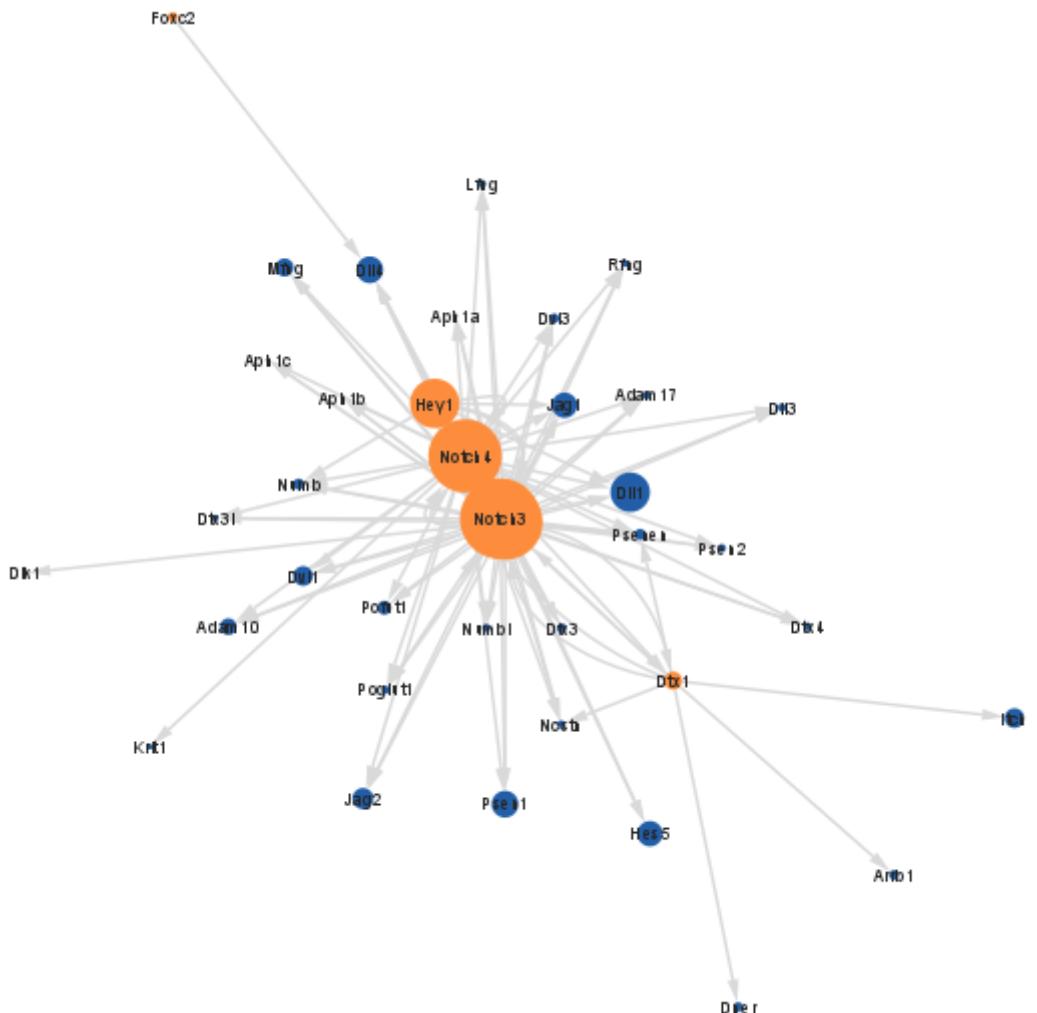
- Sphingolipid metabolism



Si tratta di un importante pathway cellulare (insieme di reazioni chimiche coinvolte nei processi catabolici e anabolici) che mette insieme diversi altri pathway, di cui la famiglia delle ceramidi prendono un ruolo centrale. In particolare i geni *Lass\_*, che si vedono svolgere una funzione centrale della community, fanno parte di questa famiglia genetica, e si tratta di geni soppressori di metastasi nei tumori umani. In particolare la presenza di *Lass2*, il più centrale, è stata individuata in carcinomi al fegato.

È la community più assortativa della rete, con un valore pari a +0.66, e allo stesso tempo quella con likelihood di riconoscimento più alto, con 0.559.

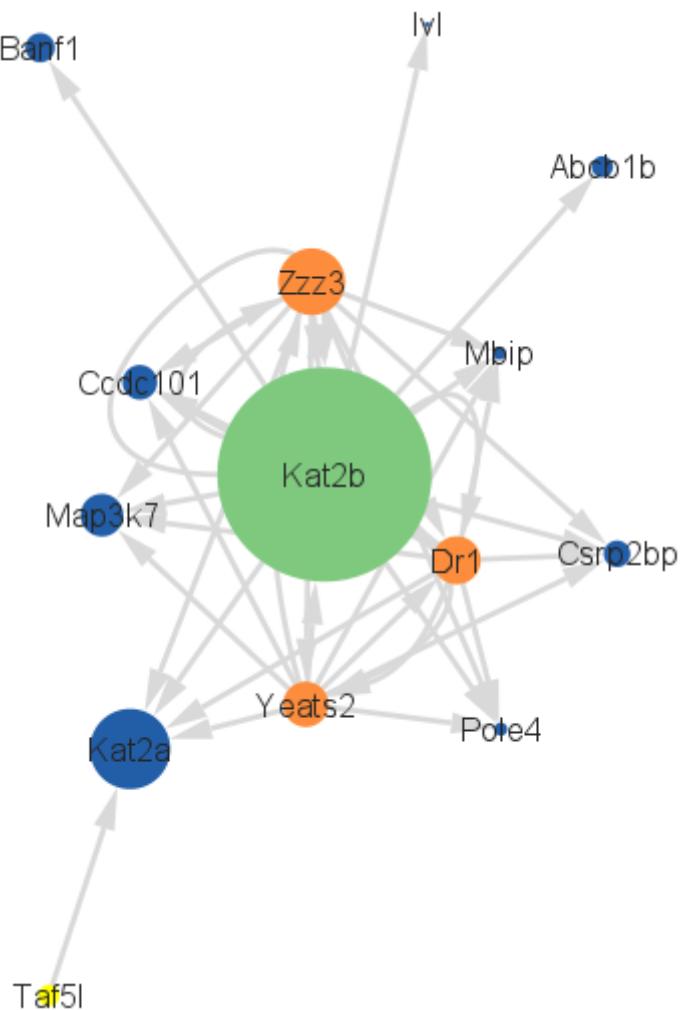
- Notch Signaling Pathway



Esso è importante per la comunicazione cellula-cellula, che coinvolge i meccanismi di regolazione genica che controllano i processi di differenziazione cellulare multipla durante la vita embrionale e adulta. La segnalazione Notch ha anche un ruolo nei seguenti processi: funzione e sviluppo neuronale.

Per la nostra rete rappresenta la seconda community più assortativa, con +0.37, oltre ad essere tra quelle con likelihood di riconoscimento più alto, a 0.448.

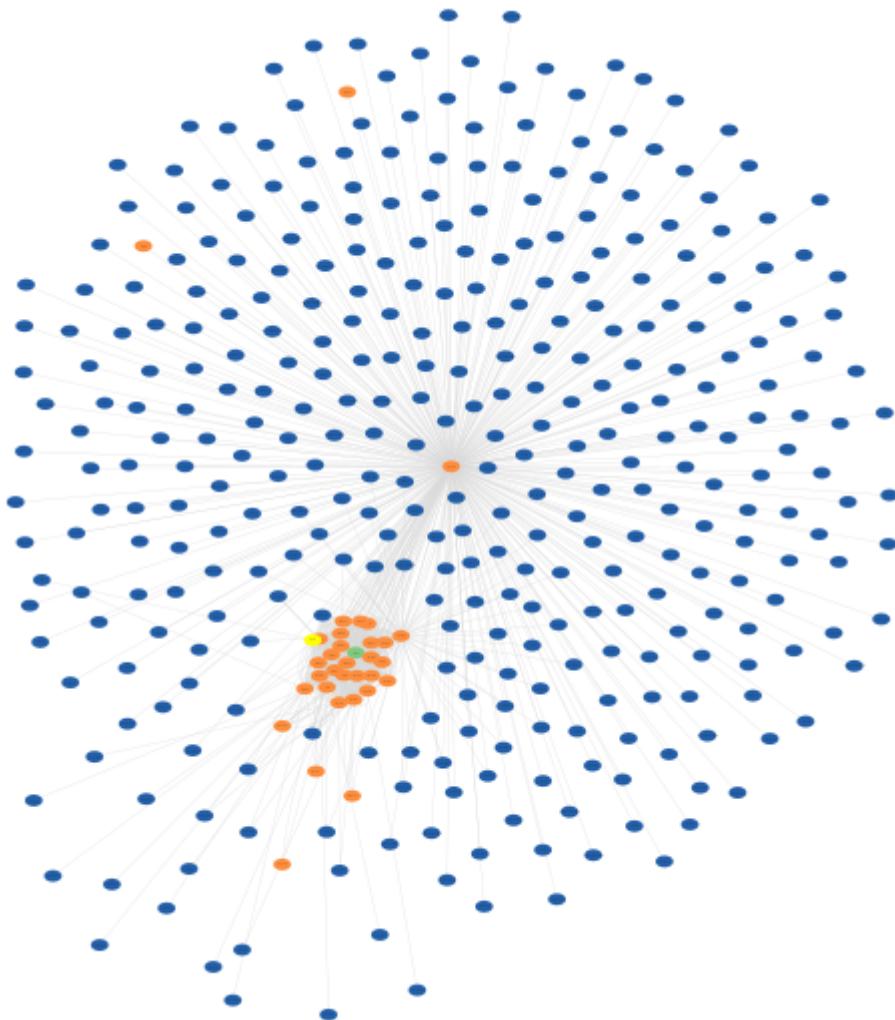
- ATAC complex



Regola attivamente la trascrizione dei geni e facilita le funzionalità dei complessi di rimodellazione della cromatina (DNA compatta in strutture dette cromosomi).

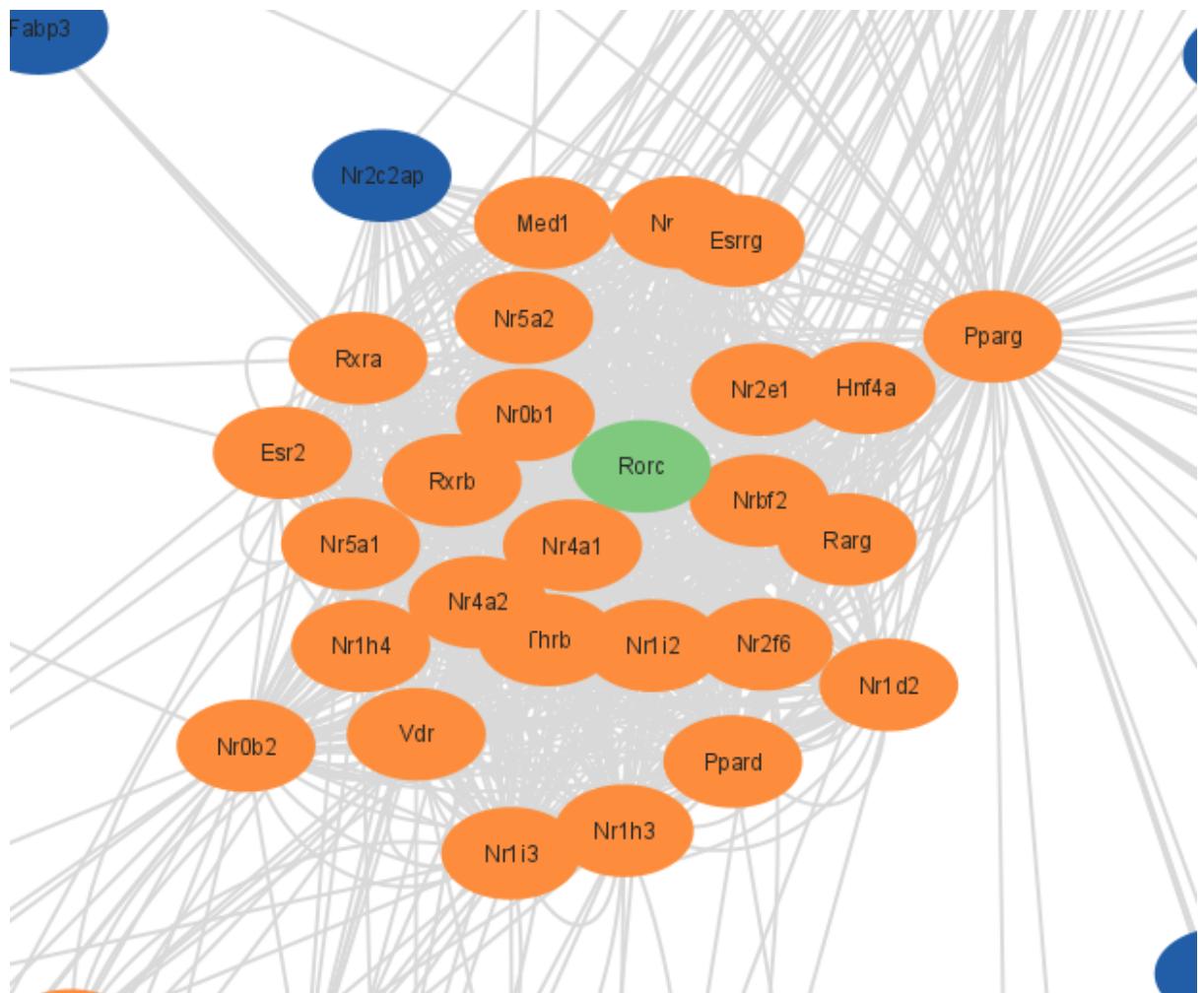
È la seconda community con maggior likelihood di riconoscimento. Il gene più centrale, *Kat2b*, TF di self-regulation, è coinvolto in processi di crescita e differenziazione della cellula.

- Nuclear Receptor transcription pathway

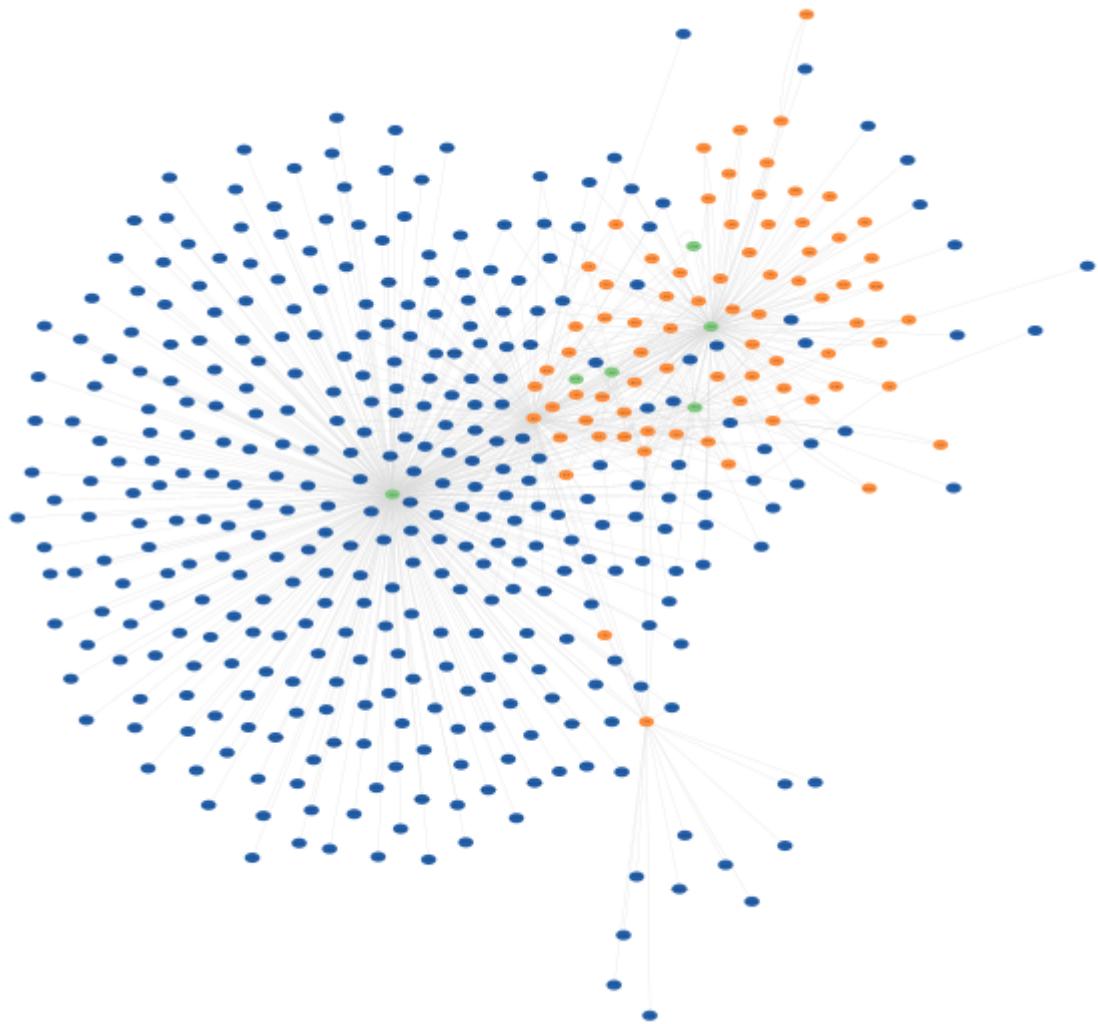


È una famiglia di geni che gioca un ruolo fondamentale nella differenziazione e sviluppo della cellula, la proliferazione (la divisione della cellula) e il metabolismo. È associata a numerose patologie tumorali, cardiovascolari e infiammazioni.

Presenta un nodo centrale, *Esrrb*, tra quelli con maggior betweenness centrality dell'intera rete, a cui sono collegati quasi tutti i target, e un cluster molto fitto di TF, nella figura che segue, di cui la maggior parte si distinguono per il prefisso *Nr*, che sta proprio per nuclear receptor, il che testimonia la correttezza nel riconoscimento del cluster attraverso il functional enrichment. Questo cluster presenta una gran frequenza di feedback loop, motivo per il quale la community è stata selezionata.

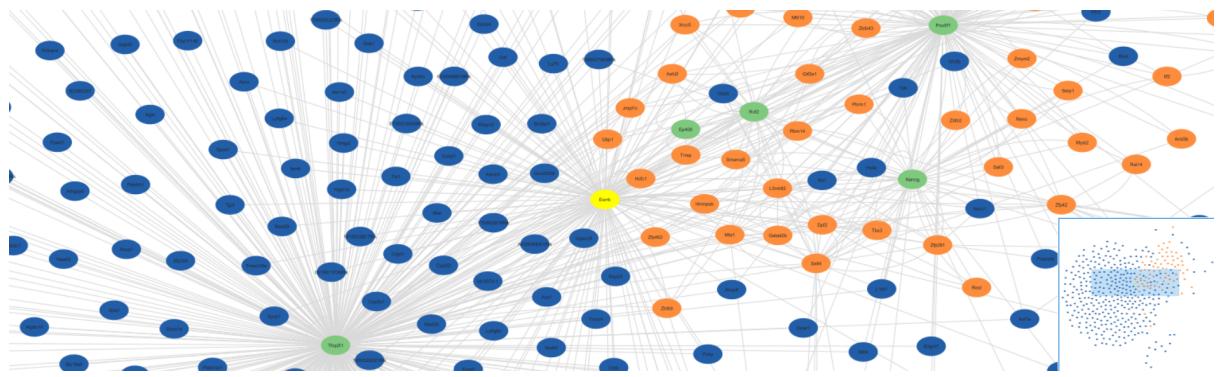


- PluriNetWork: mechanisms associated with Pluripotency

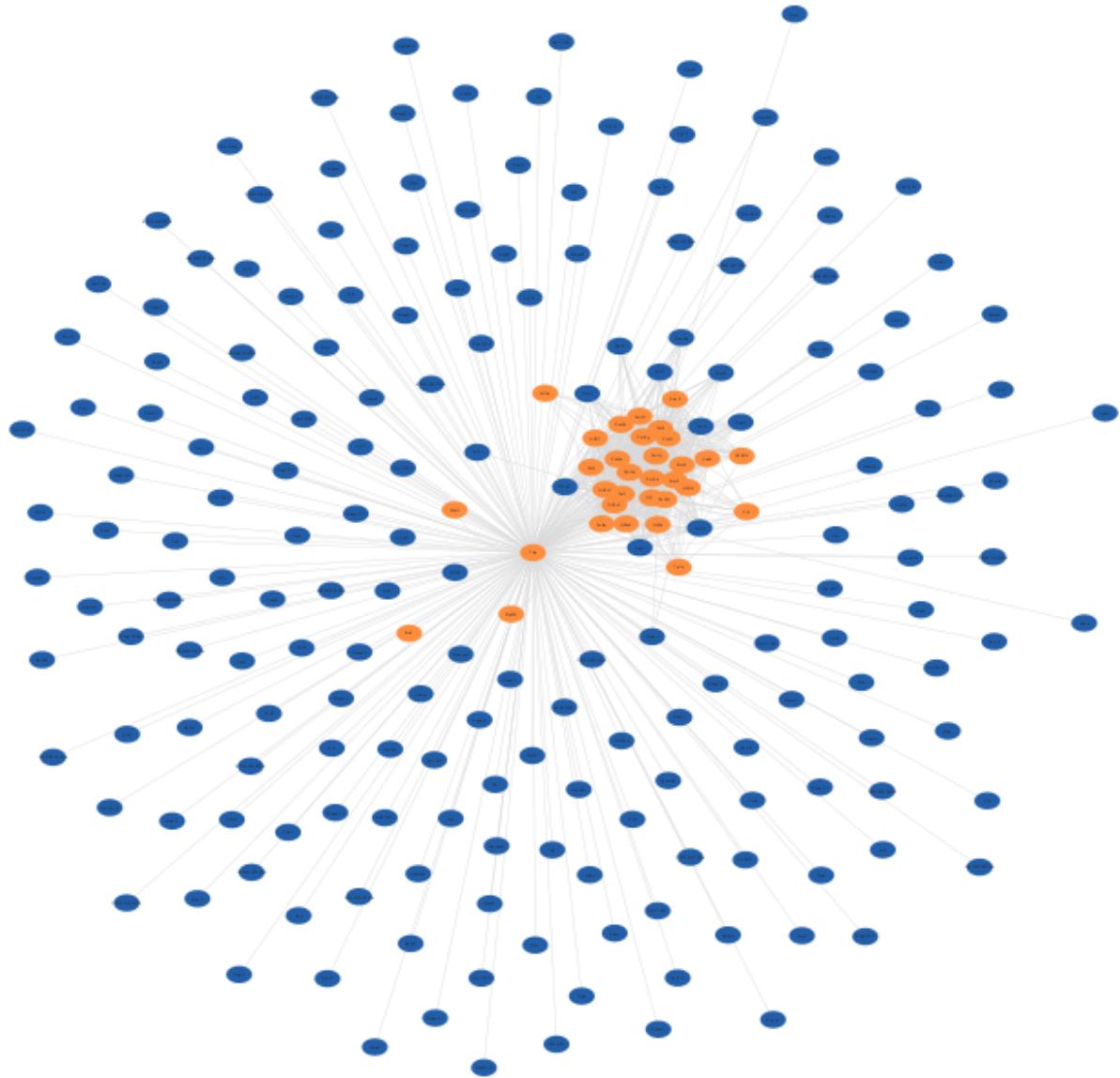


Si tratta di una delle community con più hub della rete, e racchiude quei meccanismi fondamentali per la pluripotenza, che descrive l'abilità di produrre risposte diverse a stimoli esterni in modo da conferire la capacità ad una cellula di differenziarsi in tipi cellulari differenti.

La community è costituita principalmente da 2 clusters, il primo costituito principalmente da target gene, il secondo da TF, e nella figura che segue si nota come essi siano rispettivamente dominati dai TF di self-regulation *Tfpc2l1* e *Pou5f1*, con *Esrrb* che svolge un ruolo da vero e proprio ponte tra le due, tenendole coese. Questi geni sono già stati ritrovati nell'analisi delle centralità sull'intera rete, e confermano così la loro importanza anche a livello delle singole community.



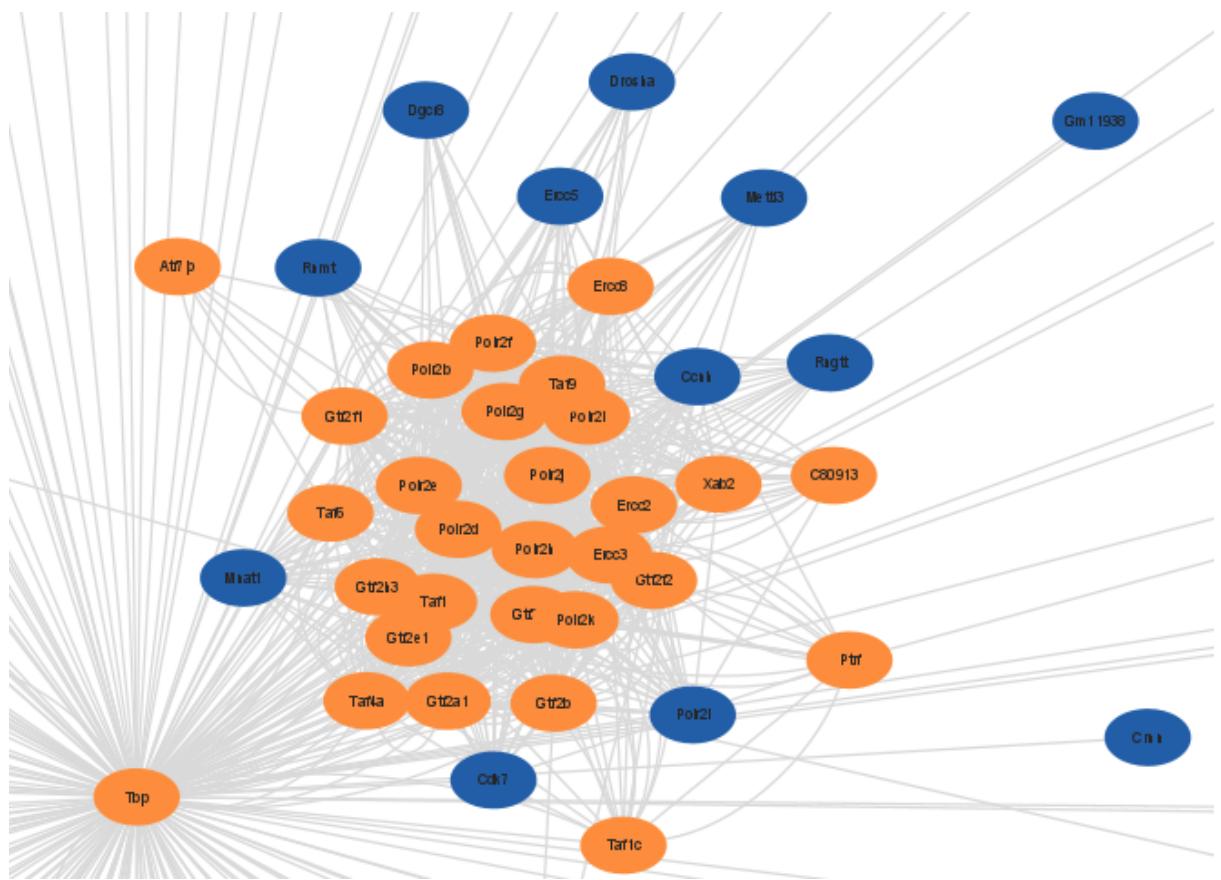
- RNA Polymerase II Promoter Escape



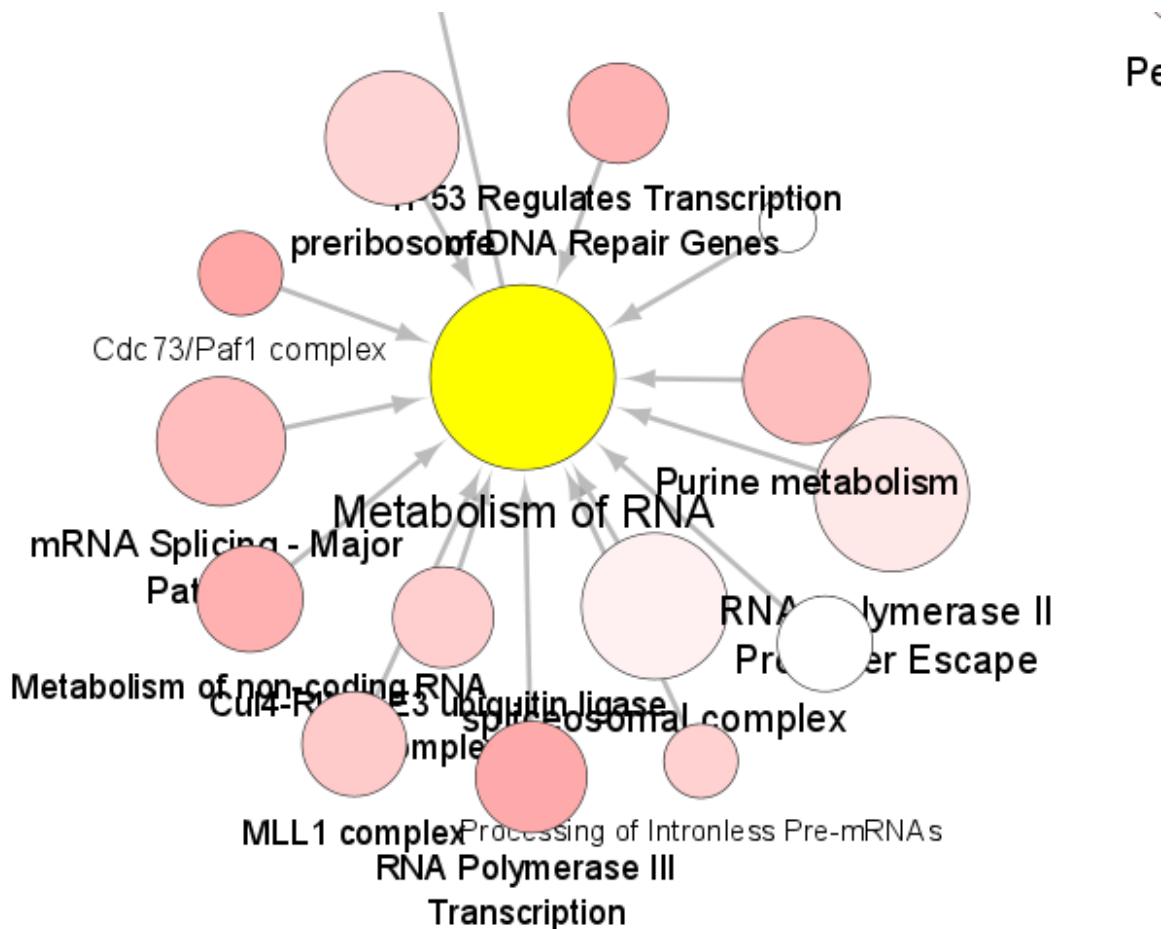
Questa community è responsabile del metabolismo dell'RNA polimerasi ed un promotore è una regione di DNA costituita da specifiche sequenze dette consenso, alla quale si lega la RNA polimerasi per iniziare la trascrizione di un gene, o di più geni. Qui ritroviamo i geni con prefisso *Polr2*, che come visto in precedenza erano quelli con eigenvector più alto dell'intera rete, e nella Giant Component fortemente connessa costituivano un grafo completo.

La community presenta una struttura molto simile a quella del *Nuclear Receptor transcription pathway*, con un subcluster molto fitto di TF che svolgono funzioni comuni.

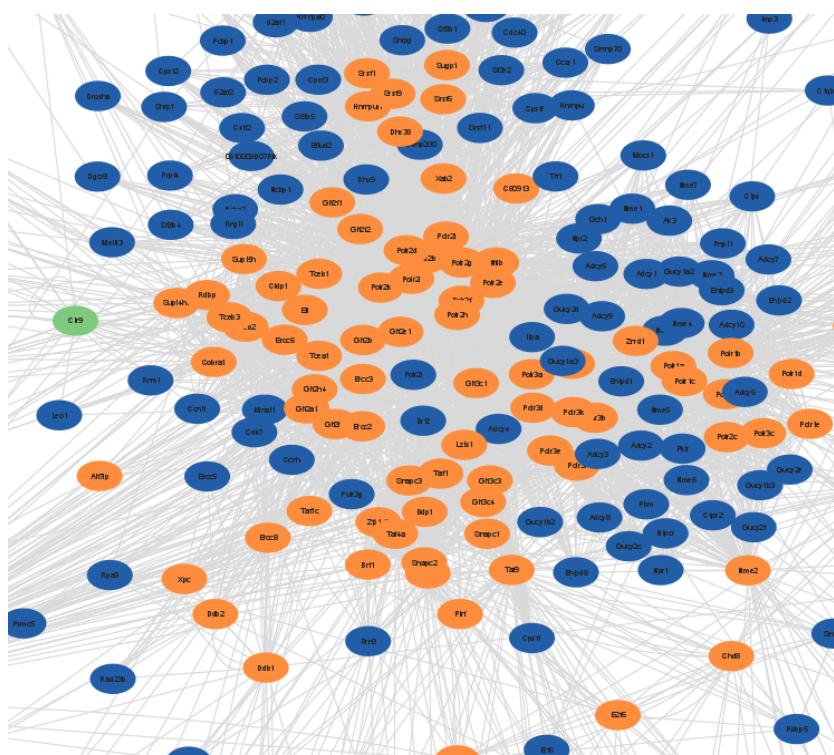
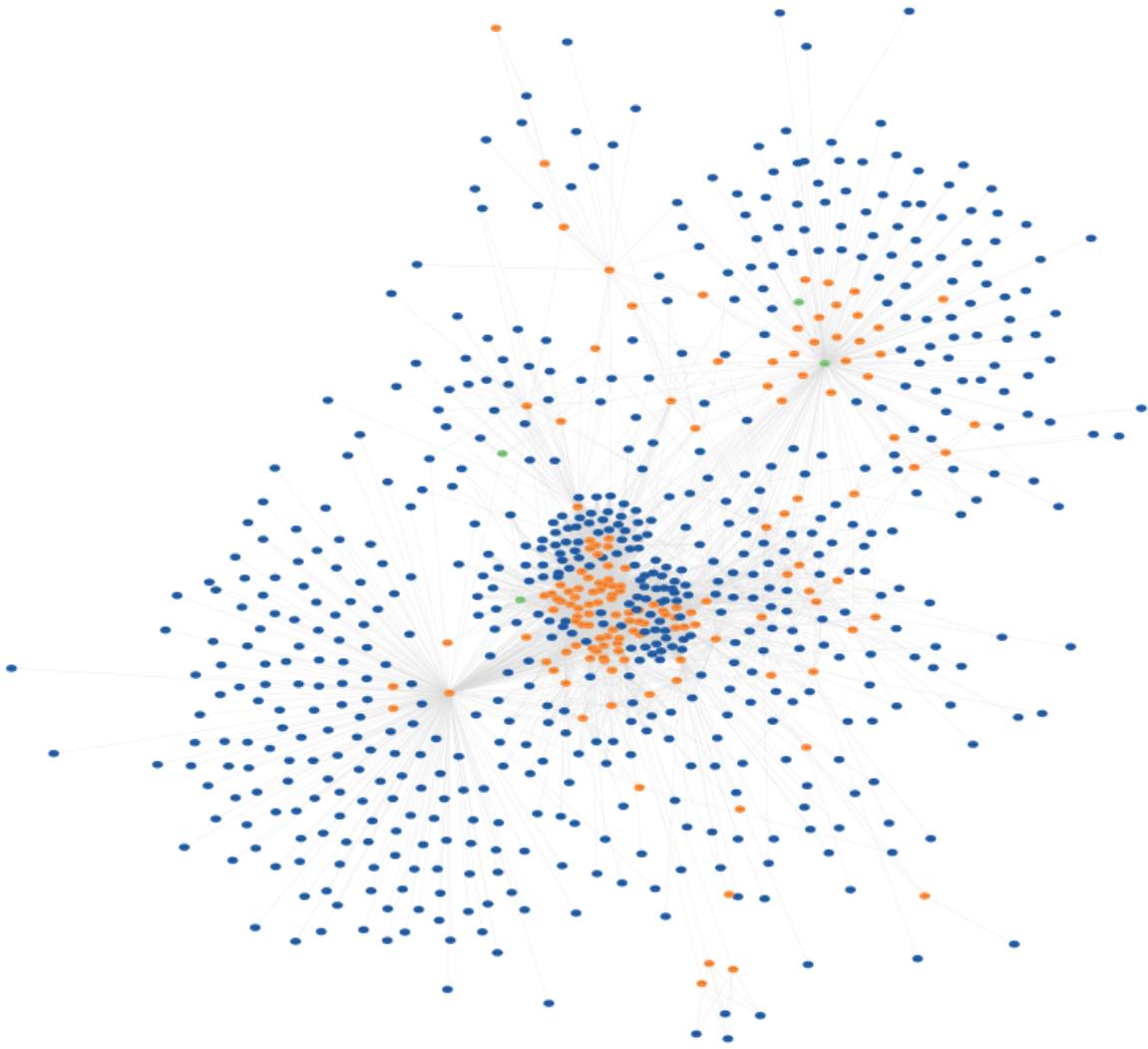
Da notare anche l'importanza del TF *Tbp*, gene centrale della community che si lega a gran parte dei target in essa contenuti.



- Metabolism of RNA

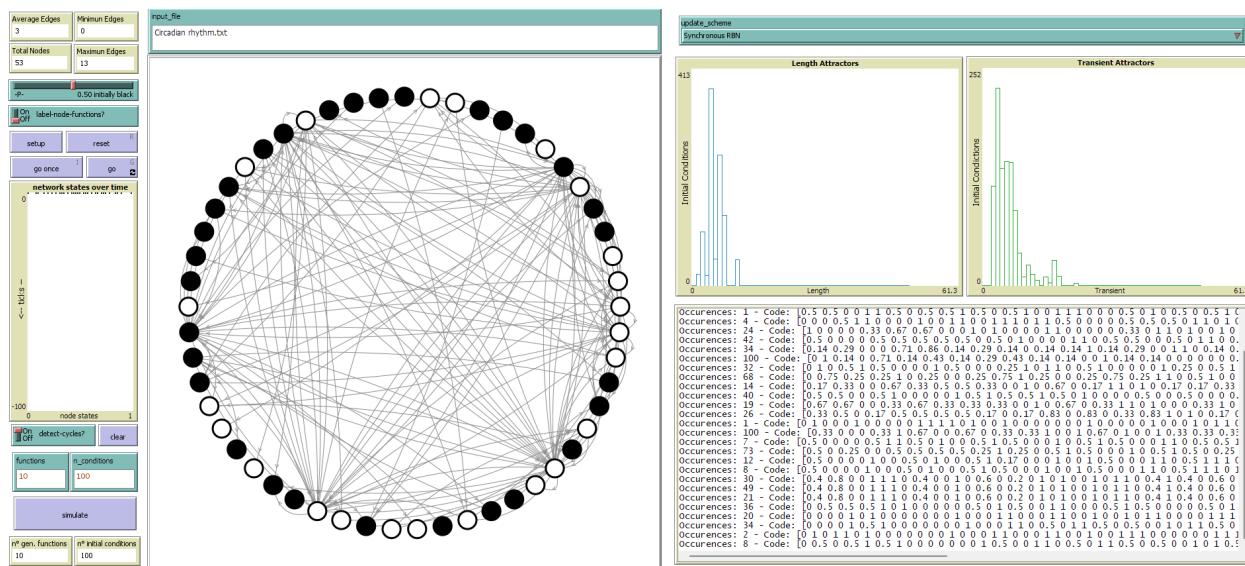


Importante riportare anche la community responsabile dell'intero metabolismo dell'RNA (formazione e degradazione), di cui fa parte anche *RNA Polymerase II Promoter Escape*, nella quale la porzione di geni *Polr2* svolge un ruolo ancora più centrale, dominando l'intero cluster centrale, il più denso, e mantenendo compatti i subcluster periferici della community.



# Proprietà dinamiche Community

Per il progetto di Simulazione io ed il mio collega abbiamo impiegato ed arricchito un modello di Random Boolean Network per analizzare dal punto di vista dinamico le community trovate tramite l'algoritmo OSLOM. Mostrerò l'esempio della community del Ritmo Circadiano. Ogni arco esprime una relazione di attivazione o disattivazione verso un determinato nodo, ma non sappiamo esattamente quando essi si andranno a spegnere o accendere; allora andremo a simulare questa rete rendendo dinamico questo meccanismo. Ogni nodo avrà una funzione booleana casuale associata che li farà cambiare stato (acceso o spento) a seconda dello stato di ingresso del suo vicinato. Partiamo nel mostrare la UI del modello caricando la community del Ritmo Circadiano tramite una matrice di adiacenza:



Nella UI si può trovare la community caricata come modello di Random Boolean Network dove potremo simulare la rete potendo selezionare su quante condizioni iniziali casuali e funzioni booleane vogliamo iterare.

Sulla destra invece si troveranno i vari grafici riassuntivi indicanti:

- Il Bar Plot delle varie lunghezze degli attrattori trovati
- Il Bar Plot del transiente dei vari attrattori trovati
- La lista di occorrenze degli attrattori con il loro relativo codice identificativo

L'analisi del comportamento dinamico che si andrà ad effettuare sarà quella di studiare in quali vari stati stazionari (*gli attrattori*), la mia Random Boolean Network è andata a convergere, dopo un tempo, indicato dal transiente, associato ad ogni attrattore che avrà una certa lunghezza. Studiando questi aspetti possiamo individuare se il gruppo funzionale esibisce un

comportamento caotico o regolare. In questo caso avremo un comportamento **regolare** perché la rete va quasi subito a convergenza: se si guardano i valori dei transienti trovati, della lunghezza degli attrattori e il numero di attrattori totali, non avremo numeri troppo alti che sarebbero tipici di una rete caotica. Intuitivamente ci potevamo aspettare questo comportamento perché il ritmo circadiano si caratterizza per essere un complesso sistema interno responsabile di cicli riguardanti la pressione arteriosa, la temperatura del corpo, il tono muscolare, la frequenza cardiaca, il ritmo sonno-veglia.... Il quale ha un comportamento ciclico e quindi regolare!

Si potrebbe pensare di unire queste conclusioni sul comportamento dinamico e risultati sugli aspetti topologici della rete che abbiamo trattato in questa relazione: ad esempio, come in questo caso, se la rete avesse un distribuzione di in-degree con dei picchi verso valori di in-degree alti, possiamo dedurre che la rete andrà a convergenza abbastanza presto e quindi presenterà un comportamento regolare. Può essere dedotto dal fatto che i nodi cambiano stato in base agli input ricevuti dai nodi adiacenti (modulando il proprio valore tramite la funzione booleana a loro assegnata) e più la rete è fitta di connessione entranti e meno tempo ci metterà ad incontrare una condizione stabile.

## Conclusioni

Il progetto ci ha portato a scontrarci con le difficoltà imposte dalle limitazioni in termini di risorse di calcolo, ma si può dire che i risultati ottenuti sono comunque soddisfacenti, in quanto le alternative adottate ai problemi che sono sorti hanno restituito risultati esaurienti.

Nonostante la bassa conoscenza del dominio, siamo comunque riusciti a dare interpretazioni pratiche ai risultati ottenuti, senza entrare troppo nel dettaglio dei concetti biologici non di nostra competenza.

L'analisi della rete ha portato a risultati rilevanti, facendoci trovare particolarità nella distribuzione dei nodi e le loro relazioni con altri geni coinvolti in funzionalità comuni.

È stato inoltre convincente l'analisi fatta sulle diverse categorie di TF, soprattutto i *self-regulation*, e l'analisi dei *regulatory circuits*, che ci hanno confermato il loro ruolo centrale nella rete e la loro capacità di conferire robustezza alla rete ed essere coinvolti in funzionalità fondamentali per l'organismo.

Le analisi effettuate attraverso la rimozione di nodi significativi ha portato a risultati interessanti, mostrando come reagirebbe la rete al danneggiamento di geni e TF.

L'algoritmo di functional enrichment non è riuscito a riconoscere la totalità delle community come si sperava, e ciò è da imputare ad una parziale base di conoscenza, o alla suddivisione stessa operata dagli algoritmi di detection. Questo ci ha portato a svolgere le analisi su di una ground truth parziale ed infatti le conclusioni sulle community ne hanno subito l'influenza.

Analizzando l'overlap dei nodi tra le community ottenute, ci aspettavamo magari una maggiore correlazione con tutte le misure di centralità, mentre siamo riusciti ad individuarla solo per quanto riguarda la *closeness*.

In quanto alla parte di analisi delle proprietà dinamiche delle community, il modello utilizzato non riesce a gestire reti di nodi con in-degree troppo alta, in quanto la simulazione ha complessità esponenziale nella generazione di funzioni booleane da associare ad ogni nodo.

# Riferimenti

[https://it.wikipedia.org/wiki/Espressione\\_genica](https://it.wikipedia.org/wiki/Espressione_genica) [definizione di RNG]

<http://www.regnetworkweb.org> [database]

<http://manual.cytoscape.org/en/stable/> [Cytoscape Documentation]

<https://www.ncbi.nlm.nih.gov/gene> [informazioni sui geni]