Università degli Studi di Padova
Corso di Laurea Magistrale in Ingegneria Informatica

Progetto di *Algoritmi per la Bioinformatica*

Network Alignment

- PPI: Protein-Protein Interaction -

Studenti: *Luca Masiero*
Stefano Ivancich

Supervisor: *Prof. Matteo Comin*

Anno Accademico 2019-2020
10 Giugno 2020

Indice

1 Introduzione: *Network Alignment* e *PPIN*

L'obiettivo del ***Network Alignment*** (traducibile con *allineamento delle reti*) consiste nel trovare somiglianze tra la struttura e/o la topologia di due o più reti.

Nel contesto biologico, confrontare le reti di diversi organismi (rappresentate tramite *grafi*) è, attualmente, uno dei problemi più importanti ed interessanti della Biologia. Gli allineamenti di reti biologiche possono infatti risultare molto utili perché, avendo molte informazioni su alcuni nodi di una determinata rete G_1 e quasi nulla su nodi topologicamente simili in un'altra G_2, la conoscenza specialistica di uno di quei nodi può dirci qualcosa di nuovo sul corrispettivo. Gli allineamenti delle reti possono anche essere utilizzati per misurare la somiglianza globale tra reti complete di specie diverse.

Le ***Protein-Protein Interaction Networks*** (**PPIN**, *reti di interazione proteina-proteina*) sono strumenti validi per comprendere le funzioni delle cellule, le malattie umane e il design e riposizionamento dei farmaci[1], nonché per ottenere una descrizione completa degli *interattomi*[2] affinché sia possibile capire, tramite la loro analisi comparativa, più profondamente i processi biologici.

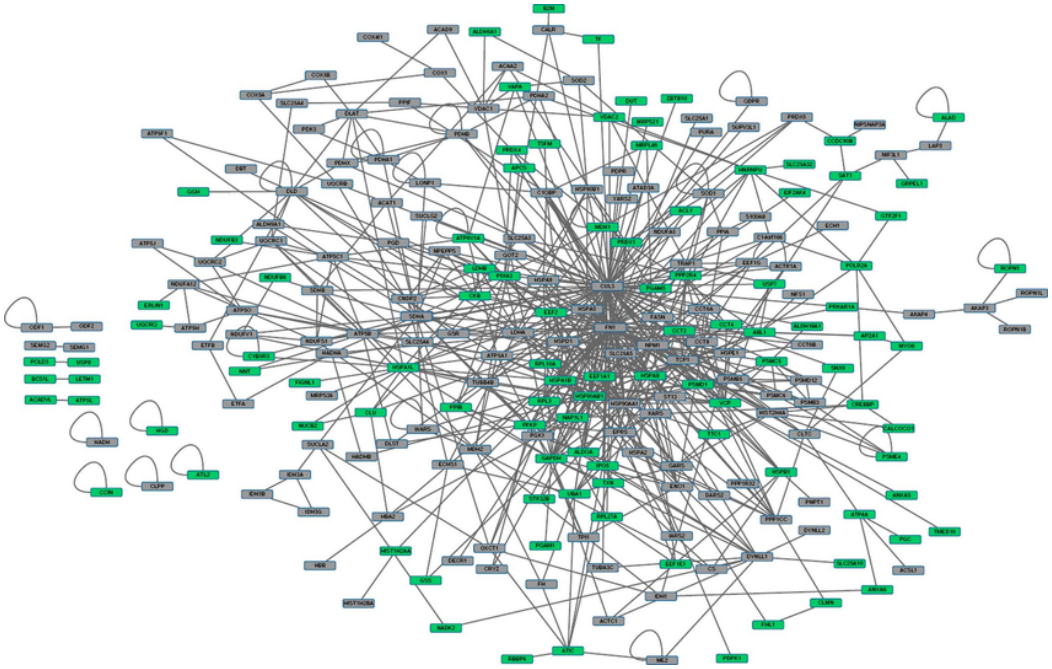


Figura 1.1*: Un esempio di PPIN.*

Interpretare una PPI è un compito particolarmente impegnativo a causa della complessità della rete. Negli anni, sono stati proposti diversi algoritmi per l'interpretazione automatica delle PPI, in un primo momento considerando esclusivamente la *topologia della rete*[3], e successivamente integrando i termini dell'*Ontologia Genica*[4] (GO) come attributi di somiglianza dei nodi.

[1] Il *drug-repositioning* è l'insieme delle analisi volte a stabilire se un farmaco già noto possa essere utilizzato per il trattamento di sintomatologie diverse da quelle descritte in etichetta.

[2] Si veda la ***Sottosezione 2.1.***

[3] La topologia di rete è il modello (grafo) finalizzato a rappresentare le relazioni di connettività, fisica o logica, tra gli elementi costituenti la rete stessa (i *nodi*).

[4] Nato nel 1988, *Gene Ontology* è un progetto bioinformatico atto ad unificare la descrizione delle caratteristiche dei prodotti dei geni in tutte le specie. In particolare, il progetto si propone di:

Negli ultimi anni, la crescente quantità e qualità dei dati *omici*[5] ha portato all'assemblaggio di reti biologiche, il cui obiettivo finale è quello di svelare i processi cellulari sottostanti. In questo scenario, le PPI sono tra le reti più importanti ed ampiamente studiate. Nelle reti PPI, un sistema biologico è descritto in termini di ***proteine***[6], che costituiscono i *nodi* del grafo, e le loro relazioni (interazioni fisico/funzionali), rappresentate dagli *archi* del grafo.

Date le grandi dimensioni (tipicamente vengono coinvolte migliaia di elementi), le reti PPI sono analizzate tramite l'identificazione di *sottoreti*, o *moduli*, che mostrano specifiche caratteristiche topologiche e/o funzionali.

L'espressione **modulo topologico** si riferisce ad un *gruppo di nodi che hanno molte più connessioni con i nodi del gruppo piuttosto che con quelli esterni.*

L'espressione **modulo funzionale** si riferisce ad un *gruppo di nodi che condividono una funzione biologica.*

Si noti che un gruppo di nodi che rappresenta un modulo può avere sia proprietà topologiche che funzionali. Idealmente, i moduli topologici e funzionali coinciderebbero; in pratica, essi costituiscono due entità diverse, anche se tipicamente si sovrappongono in larga misura. Di conseguenza, sia la topologia della rete che le informazioni funzionali contribuiscono alla comprensione complessiva dei meccanismi biologici delle reti PPI.

Nella prossime sezioni presenteremo gli argomenti necessari per comprendere il funzionamento dei metodi attualmente più efficienti per l'analisi delle PPIN.

1. Mantenere e sviluppare un vocabolario controllato atto a descrivere i geni e i prodotti genici per ogni organismo vivente;
2. Annotare i geni e i prodotti genici, e diffondere tali dati;
3. Fornire strumenti per un facile accesso ai dati forniti dal progetto.

[5] Quando si parla di "scienze omiche" si intendono quelle discipline che hanno per oggetto di studio l'insieme di geni (genomica), dei trascritti (trascrittomica), delle proteine (proteomica) e dei metaboliti (metabolomica) che vengono espressi da una cellula, diversamente da quanto fanno le scienze biologiche tradizionali che invece si occupano di studiare i processi biologici singolarmente. Si tratta, dunque, di guardare cellule e tessuti da una prospettiva diversa, prospettiva che probabilmente meglio si addice a descrivere dei sistemi come quelli biologici caratterizzati da un elevato grado di complessità.

[6] Le proteine sono macromolecole biologiche costituite da catene di amminoacidi legati uno all'altro da un legame peptidico (ovvero un legame tra il gruppo amminico di un amminoacido e il gruppo carbossilico dell'altro amminoacido, creato attraverso una reazione di condensazione con perdita di una molecola d'acqua). Le proteine svolgono una vasta gamma di funzioni all'interno degli organismi viventi, tra cui la catalisi delle reazioni metaboliche, funzione di sintesi (come la replicazione del DNA), la risposta agli stimoli e il trasporto di molecole da un luogo ad un altro. Le proteine differiscono l'una dall'altra soprattutto nella loro sequenza di amminoacidi, la quale è dettata dalla sequenza nucleotidica conservata nei geni e che di solito si traduce in un ripiegamento proteico e in una struttura tridimensionale specifica che determina la sua attività.

2 *Protein-Protein Interaction Networks*

Le ***interazioni proteina-proteina*** (PPI) sono essenziali per quasi tutti i processi che avvengono all'interno di una cellula; comprendere a pieno queste interazioni è a sua volta fondamentale per studiare la fisiologia cellulare in condizioni normali o di malattia, o per sviluppare nuovi farmaci (dato che possono influenzare le PPI stesse). Le PPIN sono rappresentazioni matematiche dei contatti fisici tra le proteine all'interno della cellula. Questi contatti:

- sono specifici;
- si verificano tra le regioni di legame (*binding regions*) nelle proteine;
- hanno un particolare significato biologico (cioè svolgono una funzione specifica).

Le informazioni riguardanti le PPI possono rappresentare sia le interazioni transitorie che quelle stabili, in particolare:

- Le interazioni *stabili* si formano in complessi proteici (e.g. ribosoma[7], emoglobina[8]).
- Le interazioni *transitorie* sono brevi interazioni che modificano o trasportano una proteina, portando ad ulteriori cambiamenti; costituiscono la parte più dinamica dell'**interattoma**, di cui parleremo fra un attimo.

La conoscenza delle PPI può essere utilizzata per assegnare ruoli putativi[9] alle proteine non caratterizzate o caratterizzare le relazioni tra le proteine che formano complessi multimolecolari.

2.1 L'interattoma

L'***interattoma***, rappresentato tramite grafo, è *l'insieme complessivo delle interazioni molecolari in una particolare cellula* e costituisce la totalità delle PPI che si verificano all'interno della stessa, ma anche in un organismo o in un contesto biologico specifico. Lo sviluppo di tecniche di screening PPI su larga scala ha portato ad un'esplosione nella quantità di dati disponibili e la costruzione di interattomi sempre più complessi e completi (si veda la ***Figura 2.1***).

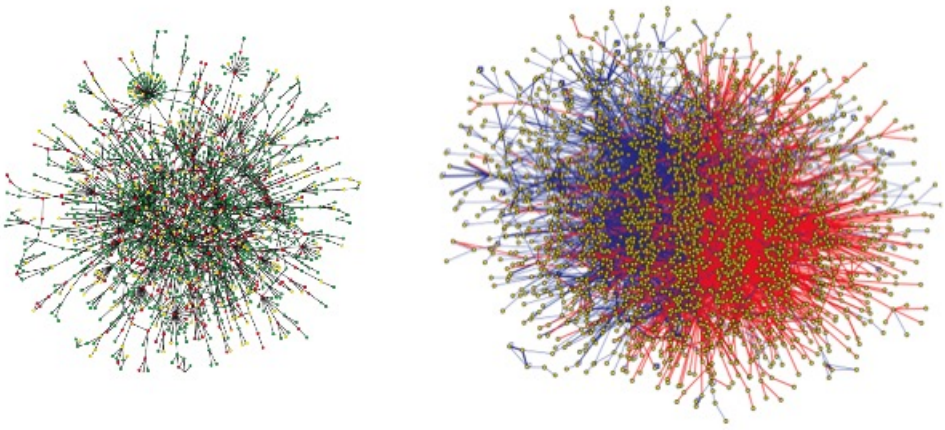


Figura 2.1*: Interattoma del lievito (sinistra) ed interattoma umano (destra).*

[7] I ribosomi sono complessi macromolecolari, immersi nel citoplasma o ancorati al reticolo endoplasmatico ruvido o contenuti in altri organuli, responsabili della sintesi proteica. La loro funzione è quella di leggere le informazioni contenute nella catena di RNA messaggero.

[8] L'emoglobina è una proteina globulare mediante la quale si compie il trasporto dell'ossigeno dai polmoni ai tessuti e dell'anidride carbonica dai tessuti ai polmoni.

[9] Presunto, apparente.

A questo punto, tuttavia, è necessario sottolineare i limiti dei dati PPI disponibili. La nostra attuale conoscenza dell'interattoma è purtroppo incompleta e rumorosa (*noisy*). I metodi di rilevamento delle PPI hanno dei limiti per quanto riguarda il numero di interazioni veramente fisiologiche che possono essere rilevate e tutti i metodi per ora realizzati ed implementati individuano sia falsi positivi che negativi.

3 PPIN: proprietà fondamentali

In questa sezione diamo uno sguardo generale ad alcune delle proprietà più importanti delle PPIN.

3.1 *Effetto del piccolo mondo*

Le *reti di interazione proteina-proteina* sono soggette all'*effetto del piccolo mondo*[10], ciò significa che intercorre una grande connettività tra le proteine (***Figura 3.1***). In altre parole, si può dire che il *diametro* della rete (= numero massimo di "passi" che separano due nodi qualsiasi) è piccolo, non importa quanto grande sia la rete[11].

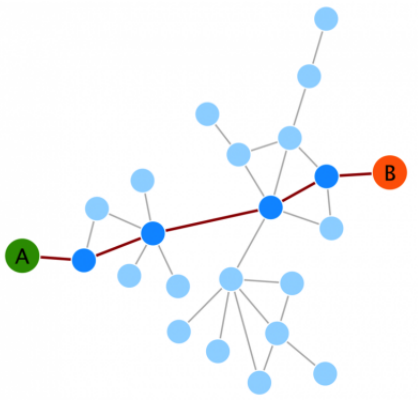


***Figura 3.1**: Effetto del piccolo mondo.*

Questo livello di connettività ha importanti conseguenze biologiche, poiché consente un flusso efficiente e rapido dei segnali all'interno della rete stessa.

A questo punto sorge spontanea una domanda: *se la rete è così strettamente connessa, perché le perturbazioni in un singolo gene o in una singola proteina non hanno conseguenze drammatiche per la rete?* I sistemi biologici sono estremamente robusti e possono far fronte a una quantità relativamente elevata di perturbazioni in singoli/e geni/proteine. Per spiegare come ciò possa accadere, dobbiamo considerare un'altra proprietà fondamentale delle PPIN, che vedremo nella prossima sottosezione.

3.2 *Scale-free networks*

Le PPIN sono ***scale-free networks***. La maggior parte dei nodi (che corrispondono alle proteine) nelle *scale-free networks* hanno solo poche connessioni con altri nodi, mentre altri (denominati *hub*) sono collegati a molti altri nodi della rete stessa (***Figura 3.2***).

[10] L'*effetto del mondo piccolo* è una teoria che sostiene che tutte le reti complesse presenti in natura sono tali che due nodi qualsiasi possono essere collegati da un percorso costituito da un numero relativamente piccolo di collegamenti.

[11] Questo, di solito, significa che due nodi sono separati da meno di sei passi (che riflettono l'ormai ampiamente diffusa teoria dei *sei gradi di separazione* usata nelle scienze sociali).

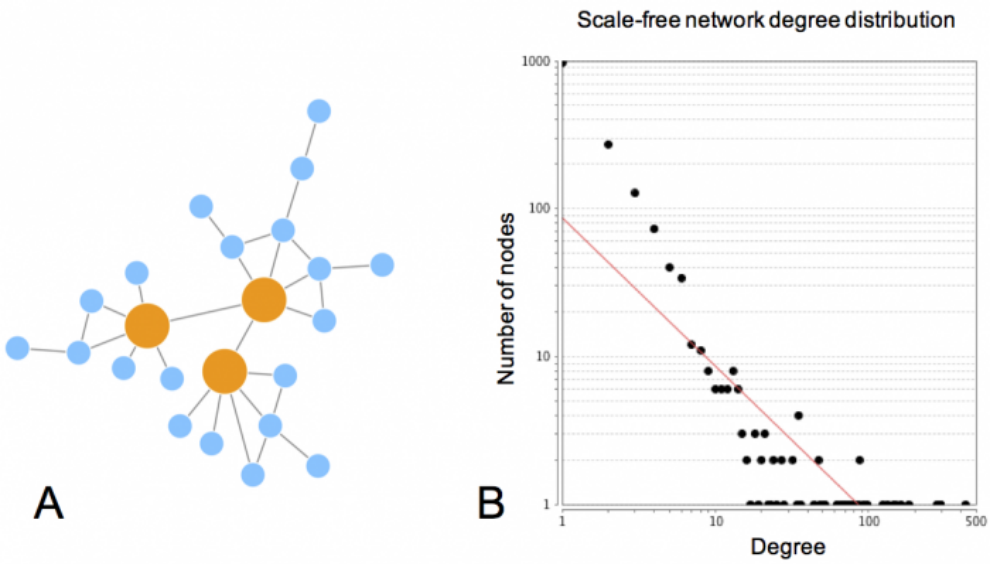


***Figura 3.2**: Il numero di connessioni di ogni nodo è chiamato **grado**. Se rappresentiamo la distribuzione del grado di una scale-free network in scala logaritmica, possiamo notare come si adatti ad una linea, avendo un piccolo numero di nodi con un alto grado (gli hub) e un grande numero di nodi con un basso grado.*

Le *scale-free networks* possono essere costruite seguendo il modello di collegamento preferenziale, noto anche come il principio "*rich get richer*" (tali reti possono essere create aggiungendo archi che sono preferibilmente collegati a nodi con un grado più elevato).

La natura *scale-free* delle reti di interazione proteina-proteina conferisce loro una serie di importanti caratteristiche:

1. **Stabilità**
   - Se i guasti si verificano in modo casuale e la maggioranza delle proteine costituisce un grado di connettività basso, la probabilità che un *hub* venga colpito è minima.
   - Se si verifica un *hub-failure*, la rete generalmente non perde la sua connettività grazie ai restanti *hub*.
2. **Invarianza ai cambiamenti di scala**
   - Non importa quanti nodi o archi abbia la rete, le sue proprietà rimangono stabili.
   - La presenza di nodi è ciò che consente l'*effetto piccolo mondo* indipendentemente dalle dimensioni della rete.
3. **Vulnerabilità agli attacchi mirati**
   - Se si perdono alcuni *hub* principali, la rete si trasforma in un insieme di grafi isolati.

3.3 Transitività

Un'altra caratteristica cruciale delle PPIN è la loro modularità[12]. La ***transitività*** o *coefficiente di clustering* di una rete misura la tendenza dei nodi a raggrupparsi. Un'alta transitività significa che la rete contiene "comunità" o gruppi di nodi che sono densamente connessi (seguendo un'analogia

[12]Caratteristica di un sistema che si compone di unità distinte (*moduli*), ognuna delle quali assolve un compito specifico ed è capace di interagire con le altre.

delle scienze sociali, "gli amici dei miei amici sono miei amici"). Nelle reti biologiche, trovare queste comunità è molto importante perché possono aiutare ad individuare **complessi proteici** (a titolo di esempio, si veda la ***Figura 3.3***).

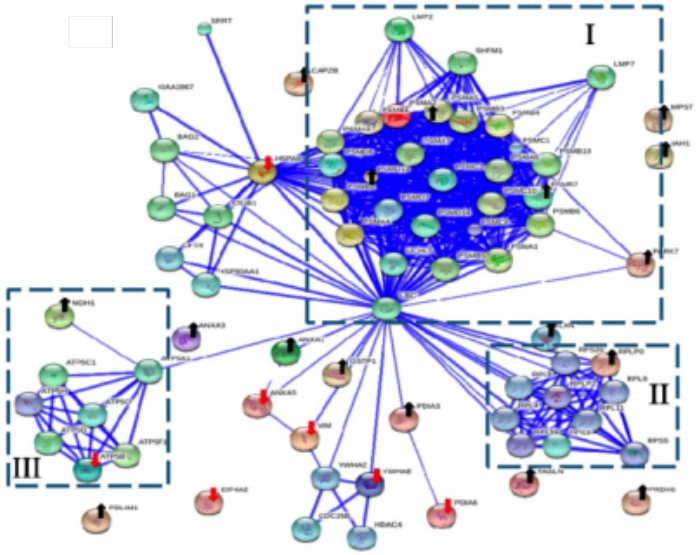


Figura 3.3*: Cluster topologici che riflettono la funzione biologica. I cluster sono evidenziati all'interno di quadrati a linee tratteggiate.*

I *complessi proteici* possono essere considerati un tipo di ***modulo*** (un'unità funzionale ed intercambiabile) in cui le proteine interagiscono in modo stabile, mantenendo una configurazione più o meno costante sia nel tempo che nello spazio.

Lo studio dei moduli è utile anche per definire le *interazioni intermodulari* tra le proteine.

4 PPIN: sorgenti di dati, affidabilità e confidenza

Il primo passo per eseguire l'analisi delle PPIN è, naturalmente, la costruzione di una rete. Ci sono diverse fonti di dati PPI che possono essere utilizzate ed è importante essere consapevoli dei loro vantaggi e svantaggi.

Essenzialmente, è possibile ottenere dati PPI da:

1. **Il proprio lavoro sperimentale**, dove si può scegliere come i dati verranno rappresentati e memorizzati.
2. **Un database primario di PPI**. Questi database estraggono le PPI dalle prove sperimentali riportate in letteratura utilizzando un processo di cura manuale. Costituiscono i principali fornitori di dati PPI.
3. **Un database di metadati**[13] **o un database predittivo**. Queste risorse riuniscono le informazioni fornite da diversi database primari e forniscono all'utente una rappresentazione unificata dei dati. I database predittivi vanno oltre e utilizzano i set di dati prodotti in modo sperimentale per prevedere, dal punto di vista puramente computazionale, le interazioni in aree inesplorate dell'interattoma. Questi dataset, tuttavia, sono in genere più "rumorosi" di quelli provenienti da altre fonti.

Spesso sarà necessario integrare dati PPI provenienti da più fonti, poiché nessun database ha una rappresentazione completa di tutte le informazioni necessarie. Da questo fatto derivano alcune sfide interessanti, poiché diversi database utilizzano identificatori diversi e contengono diversi tipi di dati.

A questo punto sorge una preoccupazione che ha a che fare con l'analisi della rete: *ci si può "fidare" del fatto che la rete di interazione rappresenti una "reale" interazione biologica?* Dato il rumore (*noise*) insito nelle informazioni dell'interattoma, è importante essere rigorosi e attenti quando si valutano i dati delle *interazioni proteina-proteina* che si utilizzano (potremmo infatti trovarci di fronte a rindondanze ed incoerenze). È molto importante tener conto del fatto che l'interattoma potrebbe essere incompleto o frammentario, per questo motivo esistono modi diversi per accertare l'affidabilità dei dati che si stanno considerando. Alcune strategie si avvalgono dei seguenti metodi:

1. **Informazioni biologiche contestuali** riguardanti le proteine e/o le molecole che sono coinvolte nell'interazione.
2. **Contare quante volte una data interazione è stata riportata in letteratura**. Questo è un approccio molto popolare e semplice; esistono varianti più elaborate di questa strategia, come il metodo MIscore[14].
3. **Metodi aggregati** che utilizzano una serie di strategie diverse e le integrano in un unico punteggio, come INTscore[15].

[13] Informazione che descrive un insieme di dati.

[14] Per una descrizione completa e rigorosa di questo metodo rimandiamo alla lettura del seguente paper (non inserito nella **Bibliografia**): Villaveces, J.M., et al., Merging and scoring molecular interactions utilising existing community standards: tools, use-cases and a case study. Database (Oxford), 2015. 2015.

[15] A titolo informativo, il paper (non inserito nella **Bibliografia**) riguardante INTscore è il seguente: Kamburov, A., Stelzl, U., and Herwig, R. IntScore: a web tool for confidence scoring of biological interactions. Nucleic Acids Res, 2012. 40(Web Server issue): p. W140-6.

5 PPIN: analisi topologica

L'analisi delle caratteristiche topologiche di una rete è un modo utile per identificare i partecipanti e le sottostrutture rilevanti che possono avere un significato dal punto di vista biologico. Ci sono molte strategie diverse che possono essere usate per fare ciò (si veda la ***Figura 5.1***); in questa sezione ci concentreremo sull'analisi della centralità (*centrality analysis*) e sul clustering topologico (*topological clustering*).

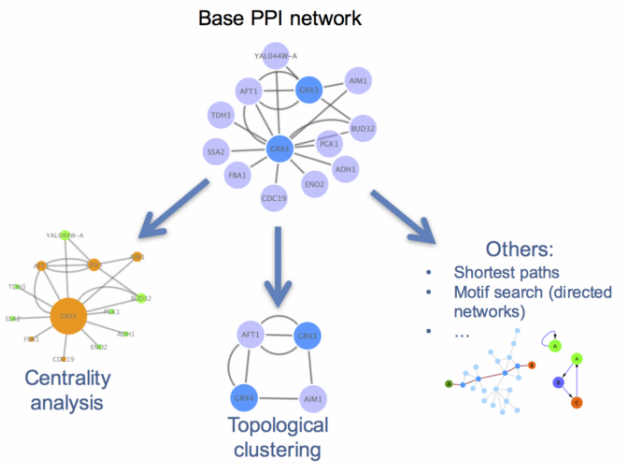


***Figura 5.1**: Strategie comuni di analisi strutturale per le PPIN.*

5.1 *Centrality Analysis*

La *centralità* fornisce una stima di quanto sia importante un nodo o un arco per la connettività della rete (***Figura 5.2***). L'analisi della centralità nelle PPIN di solito mira a rispondere alla seguente domanda: *quale proteina è la più importante e perché?*

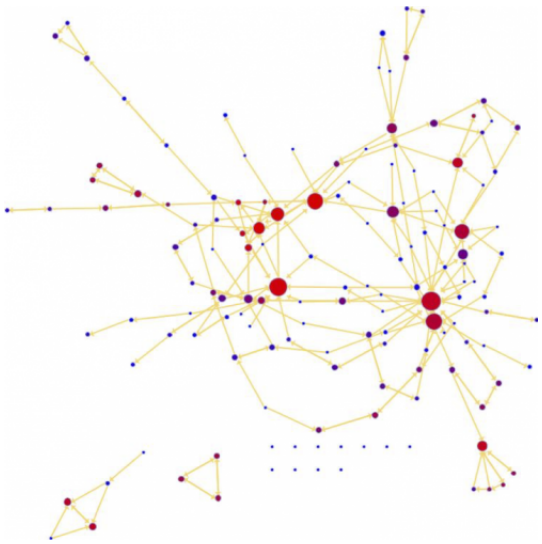


***Figura 5.2**: Centralità del nodo rappresentato in una rete. I nodi più grandi (evidenziati in rosso) hanno valori di centralità più alti.*

La definizione di *centralità* varia a seconda del contesto o dello scopo dell'analisi che si sta eseguendo e può essere misurata utilizzando diverse metriche e criteri, per esempio:

1. Il **grado dei nodi**.
2. Le **misure di centralità globale**. Due delle misure di centralità globale più utilizzate sono le centralità di *prossimità* (*closeness centrality*) e di *interrelazione* (*betweenness centrality*).

   La *closeness centrality* è una misura che stima la velocità del flusso di informazioni attraverso un dato nodo verso altri nodi. Essa misura quanto brevi sono i percorsi da un nodo verso tutti gli altri nodi.
   La *betweenness centrality* misura la frequenza con cui un nodo viene a trovarsi su tutti i percorsi più brevi fra due nodi[16].

5.2 *Clustering Analysis*

La ricerca di "comunità" all'interno di una rete è una buona strategia utile a ridurre la complessità della rete stessa e ad estrarre moduli funzionali (ad esempio, complessi proteici). I ***cluster*** sono un gruppo di nodi che sono più connessi al loro interno che con il resto della rete.

Quando si parla di PPIN, le comunità rientrano in due categorie: moduli funzionali e complessi proteici (si veda la ***Sottosezione 3.3*** per la corretta definizione dei termini).

5.2.1 Metodi di *Clustering Analysis*

Ora presentiamo brevemente due metodi che utilizzano esclusivamente la topologia della rete per individuare componenti strettamente connesse tra loro. Bisogna sottolineare che non si fanno ipotesi sulla struttura interna dei *cluster*, concentreremo pertanto la nostra attenzione soltanto sulle regioni ad alta densità.

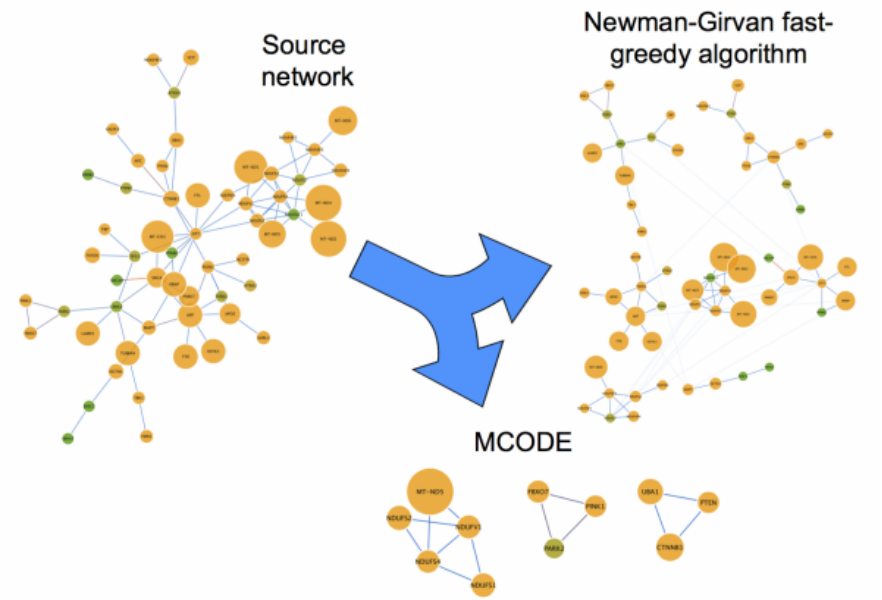


***Figura 5.3**: Metodi di Clustering Analysis.*

È importante notare che trovare la migliore struttura di "comunità" è algoritmicamente complesso ed è possibile solo per reti molto piccole. Per questo motivo sono stati sviluppati diversi metodi di approssimazione; ne presentiamo solamente due:

[16] Questi nodi possono rappresentare proteine importanti nei percorsi di segnalazione e possono rappresentare obiettivi per la scoperta di farmaci. Combinando questi dati con l'analisi delle interferenze possiamo simulare attacchi mirati alle PPIN e prevedere quali proteine sono candidate migliori per la ricerca di farmaci.

1. **Algoritmo *Newman-Girvan fast-greedy***: Questo metodo *naïve* individua i *cluster* grazie alla *edge betweenness centrality measure*. Gli archi che collegano i diversi *cluster* hanno *centrality values* più elevate. Per definire i *cluster* il metodo utilizza *edge betweenness centrality scores* per classificare gli archi della rete, quindi rimuove gli archi più centrali e ricalcola i *betweenness scores* fino a quando non rimangono più archi. Gli archi interessati dalla rimozione sono considerati parte dello stesso *cluster*.

2. ***Algoritmo MCODE***: Questo metodo è stato appositamente sviluppato per trovare complessi proteici nelle PPIN. Più rigoroso dell'algoritmo Newman-Girvan, mira a trovare solo quelle sottoreti che sono altamente interconnesse (rappresentanti complessi proteici relativamente stabili, che funzionano come una singola entità nel tempo e nello spazio). L'algoritmo utilizza un processo a tre fasi: (1) con il *weighting* un punteggio più alto viene assegnato a quei nodi i cui vicini sono più interconnessi; (2) partendo dal nodo (denominato *seed*) con il peso più elevato, vengono aggiunti al complesso i nodi che hanno un peso superiore ad una determinata soglia; (3) vengono applicati dei filtri per migliorare la qualità del *cluster*.

6 PPIN: *annotation enrichment analysis*

Ci sono molti approcci diversi che possono essere utilizzati per comprendere il contesto biologico delle PPIN. L'*annotation enrichment analysis* è uno dei metodi più popolari. Anche se non è propriamente uno strumento di analisi delle reti, è spesso utilizzato in combinazione con l'analisi topologica delle stesse.

Si utilizzano le annotazioni geniche/proteine fornite, per esempio, dall'Ontologia Genica (GO) per rispondere, attraverso un test statistico, alla seguente domanda:

"Quando si campionano X proteine (*test set*) da N proteine (*reference set*; grafo o *annotation*), qual è la probabilità che x, o più, di queste proteine appartengano ad una categoria funzionale C condivisa da n delle N proteine nel *reference set*?"

Il risultato di questo test ci fornisce una lista di termini che descrivono la rete (o una parte di essa) nel suo insieme, per identificare le "comunità" interconnesse trovate attraverso il *topological clustering*.

I principali limiti dell'*annotation enrichment analysis* derivano dalle *annotation* stesse. Alcune aree della Biologia sono annotate più approfonditamente e meglio descritte di altre, con termini più dettagliati e più precisi (nel nostro caso, solo le proteine più "popolari" sono meglio annotate). Questo introduce una certa "distorsione" nell'analisi statistica.

È anche importante notare che i termini dell'Ontologia Genica (GO) possono essere assegnati sia da un curatore umano che esegue un'attenta annotazione manuale, sia da approcci computazionali che utilizzano le basi dell'annotazione manuale per dedurre quali termini descriverebbero in modo corretto i prodotti genici non scoperti. Ne consegue che un'altra limitazione è costituita dalla complessità e dal dettaglio dell'annotazione associati a grandi insiemi di geni o proteine.

Date due reti, **allinearle** significa trovare un mapping nodo-a-nodo (= *alignment*) tra le reti in grado di ottimizzare due obiettivi: (1) massimizzare il numero di proteine mappate (corrispondenti ai nodi nel grafo) che sono correlate da un punto di vista funzionale e (2) massimizzare il numero di interazioni comuni (archi) tra le reti.

Il problema del *Network Alignment* è un problema intrattabile dovuto all'$\mathcal{NP}$-completezza sottostante al *sub-graph isomorphism problem*[17], individuato da Spephen Cook nel 1971.

7 MTGO

Il metodo MTGO - *Module detection via Topological information and GO knowledge* - costituisce un nuovo approccio di identificazione dei moduli funzionali nelle PPIN. Questo metodo combina le informazioni provenienti dalla topologia delle reti con la conoscenza biologica relativa alle proteine.

Per identificare i moduli più interessanti, MTGO utilizza partizioni ripetute della rete sfruttando la *modularità* del grafo, corrispondente ad una funzione che misura la qualità topologica di una determinata partizione in un grafo. La partizione viene successivamente appresa attraverso un processo di ottimizzazione che tiene conto della struttura della rete e della sua natura biologica. A differenza dei precedenti approcci basati su GO, MTGO fornisce un unico termine GO che descrive al meglio la natura biologica di ogni modulo identificato.

Evidenziando i principali processi coinvolti nel sistema biologico, rappresentato dai modelli di PPIN, e grazie al suo modo unico di sfruttare l'Ontologia Genica, MTGO si differenzia in maniera significativa dagli algoritmi allo stato dell'arte (ClusterOne, MCODE -***Sottosezione 5.2.1***-, COACH, CFinder, Markov Cluster MCL, DCAFP e GMFTP).

Per valutare le performance di MTGO sono state selezionate, per i test, quattro PPIN reali: Krogan, Gavin, Collins e DIP Hsapi PPIN.

	Nodes	GO-covered nodes	Edges
Krogan	2709	2537	7123
Gavin	1856	1778	7669
Collins	1622	1596	9074
Human	2734	2474	4058
Integrated	3232	3020	16948

Figura 7.1*: In questa tabella vengono indicate le caratteristiche principali di ogni rete, incluso il numero di nodi coperti dai termini GO, usati come input per MTGO.*

I termini GO utilizzati come input per MTGO includono le seguenti tre categorie: *Componente Cellulare*, *Processo Biologico* e *Funzione Molecolare*. MTGO ha mostrato i risultati migliori otto volte su nove ed è in grado di individuare complessi piccoli/sparsi anche in reti molto grandi.

MTGO è in grado di individuare moduli funzionali all'interno delle PPIN, prevede l'*overlapping* e la copertura totale della rete[18], due *features* importantissime per gli algoritmi di identificazione di moduli.

MTGO prevede una mappa sia dei moduli topologici che funzionali. I moduli topologici assicurano la copertura totale della rete, mentre quelli funzionali condividono i nodi, permettendo l'*overlapping*; il metodo dipende fortemente dalla qualità dei termini GO, le sue performance infatti

[17] Problema computazionale nel quale, dati due grafi G e H in input, si vuole determinare se G contiene un sottografo isomorfico ad H, deve cioè esistere una corrispondenza biunivoca tra gli elementi dei grafi.

[18] Si definisce *copertura di nodi* (*vertex cover*) di un grafo un insieme C di nodi con la proprietà che ogni arco nel grafo abbia almeno uno dei suoi estremi in C.

diminuiscono significativamente a seconda della perturbazione (del 25%, 50% e 75%) dei termini GO forniti ed è stato progettato per essere testato sia su reti pesate che non.

MTGO possiede l'abilità di individuare un insieme di termini GO fornendo un'interpretazione biologica significativa della PPIN, proprietà assente negli altri algoritmi allo stato dell'arte.

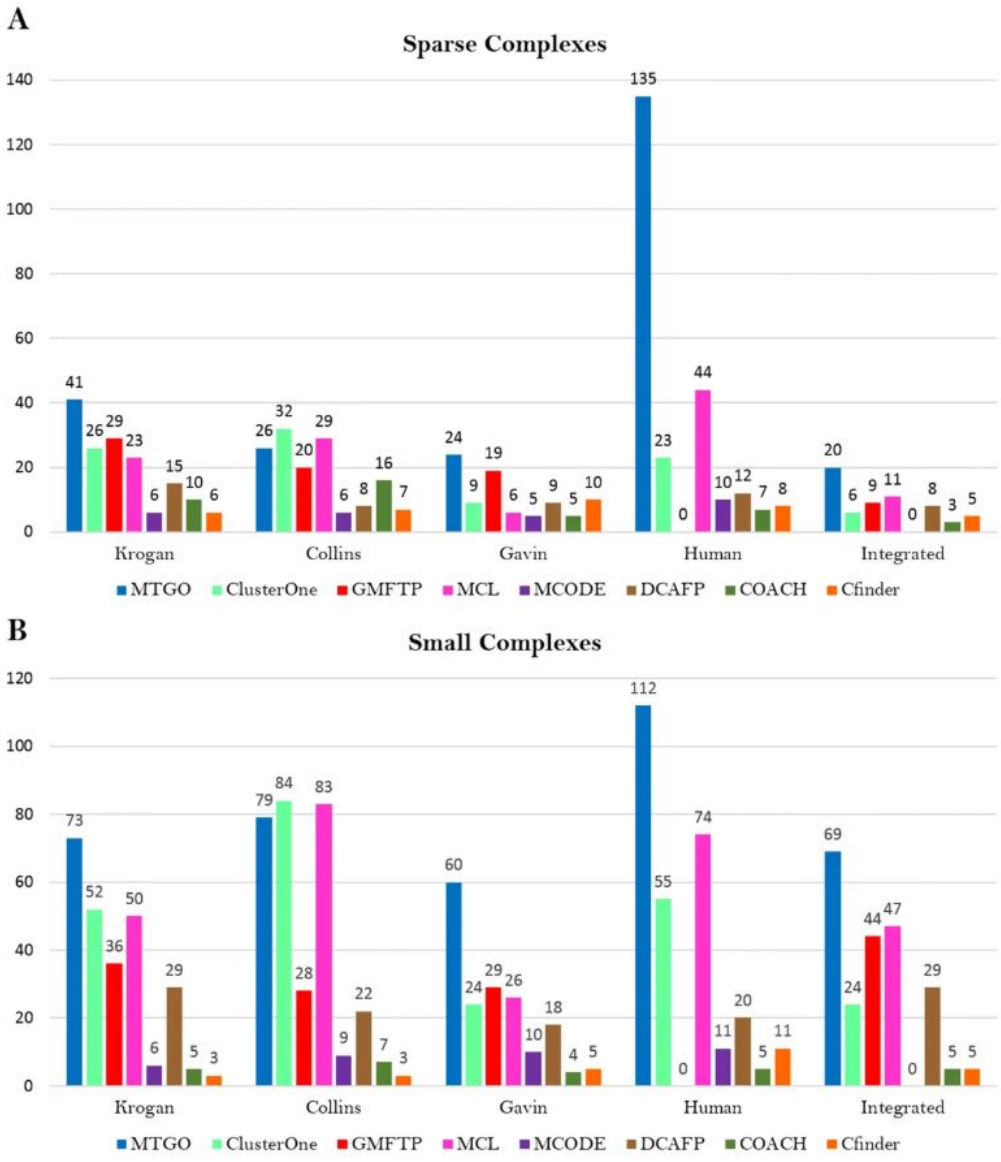


***Figura 7.2**: Confronto di MTGO con gli altri metodi.*

7.1 MTGO: inizializzazione, iterazioni e convergenza

Una PPIN può essere rappresentata tramite un grafo $G = (V, E)$ dove V ed E corrispondono ai nodi e agli archi della rete, rispettivamente. V è l'insieme delle proteine ed è definito come $V = \{v_1, v_2, v_3, ..., v_N\}$ dove N è il numero di proteine/nodi totale. E rappresenta l'insieme delle relazioni tra i nodi della rete: $E = \{e_{i,j}\}$, $(i, j) \in [1,N]$. Inoltre, G detiene le proprietà topologiche PPI. Per integrare le informazioni relative alle funzioni biologiche all'interno della rete, ai nodi vengono associati i termini GO. MTGO calcola l'insieme $T = (L, \Delta)$, dove il p-esimo elemento è

$t_p = (l_p, \delta_p)$, l_p rappresenta l'*ontology term*, mentre δ_p è l'l_p-insieme associato alla rete di proteine (si veda la ***Figura 7.3*** per un esempio).

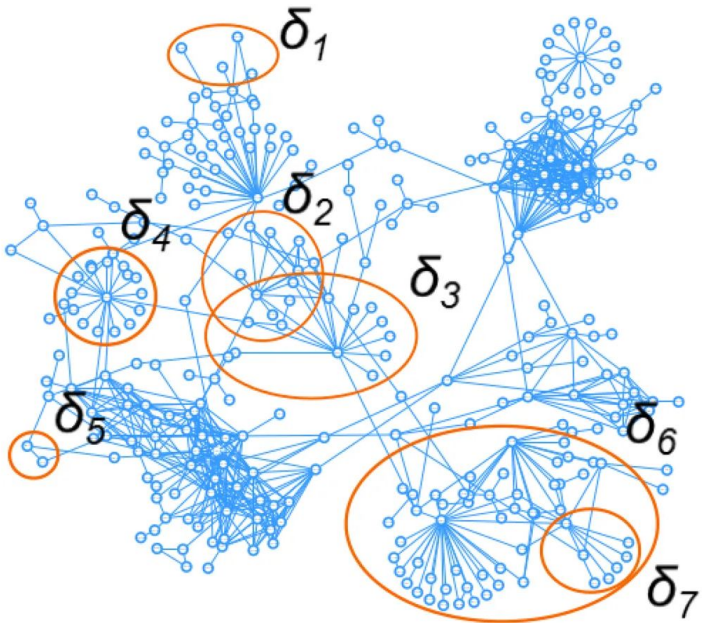


***Figura 7.3**: Esempio dei δ elementi rappresentati in una rete, potrebbero condividere più nodi o essere inclusi in una categoria più vasta.*

$I = (G, T)$ è l'input del sistema. L'obiettivo di MTGO è quello di processare G per trovare gruppi di nodi che condividono sia le proprietà topologiche (V, E), sia quelle funzionali (T). L'output di questo metodo è $R^F = (C^F, \Phi^F)$ dove C^F è l'insieme dei moduli topologici, mentre Φ^F è l'insieme dei moduli funzionali. Da notare che $|C^F| = |\Phi^F|$, la relazione è 1:1.

MTGO calcola iterativamente C e Φ e la coppia $R^F = (C^F, \Phi^F)$ viene selezionata come output finale.

L'insieme di moduli topologici C costituisce una partizione della rete, $C = \{c_1, ..., c_h, ..., c_H\}$, di modo che:

$$c_1 \cap c_2 ... \cap c_h ... \cap c_H \equiv \emptyset$$

$$c_1 \cup c_2 ... \cup c_h ... \cup c_H \equiv V.$$

Bisogna notare che ogni nodo di una partizione di C viene unicamente assegnata ad un singolo modulo topologico. D'altra parte, l'insieme $\Phi = \{\phi_1, ..., \phi_h, ..., \phi_H\}$ descrive i moduli funzionali coinvolti nella rete. Φ viene definito in questo modo:

$$\phi_1 \cap \phi_2 ... \cap \phi_h ... \cap \phi_H \equiv \emptyset$$

$$\phi_1 \cup \phi_2 ... \cup \phi_h ... \cup \phi_H \subseteq V$$

e $\Phi \subset T$, cioè Φ è il sottoinsieme di T selezionato da MTGO per descrivere le funzioni biologiche collegate alla partizione C della PPIN.

La copertura completa (*full coverage*) e la sovrapposizione (*overlapping*) sono considerate le caratteristiche ideali degli algoritmi di identificazione di moduli. MTGO garantisce entrambe queste proprietà con il suo doppio output complementare C e Φ. In particolare, i moduli topologici C

rappresentano una partizione di rete, garantendo una copertura completa per definizione. I moduli funzionali Φ si sovrappongono, consentendo l'assegnazione di un nodo a due o più moduli. Questa caratteristica è particolarmente importante in quanto riflette il comportamento dei sistemi biologici, dove una proteina può essere coinvolta in molteplici funzioni.

Vediamo ora di capire, per lo meno a livello intuitivo, i passi eseguiti da MTGO. Dato l'input $I = (G, T)$, MTGO realizza tre fasi principali: (1) inizializzazione, (2) iterazione e (3) controllo della convergenza.

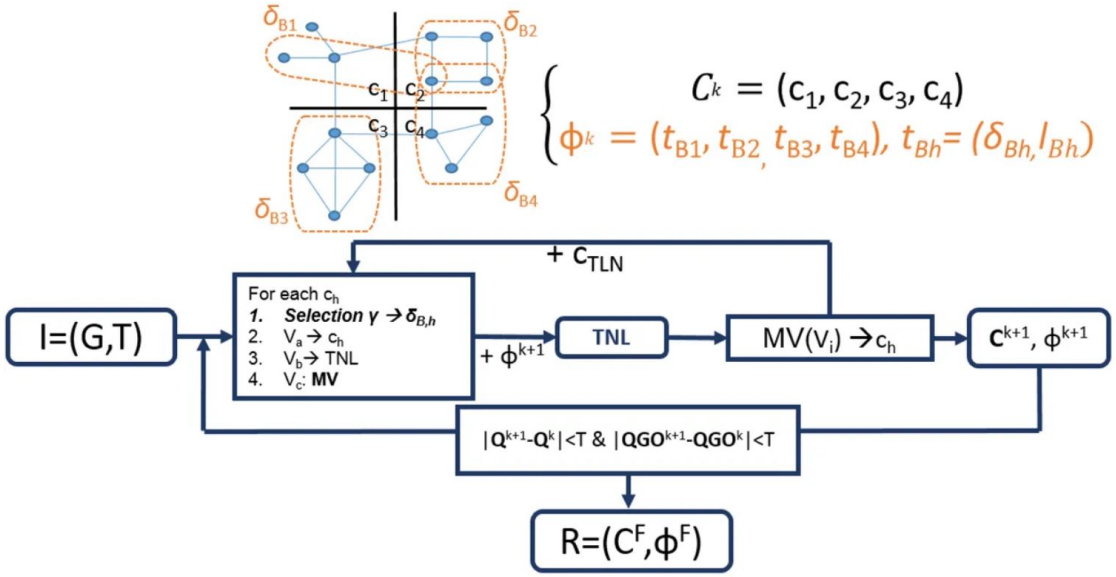


***Figura 7.4**: Rappresentazione dei passi seguiti dall'algoritmo MTGO.*

Nella fase di **inizializzazione**, V viene utilizzato per generare una partizione casuale C^0 nella quale il numero di moduli topologici è $\propto \sqrt{N}$. T viene creato partendo da una *GO term list* fornita dall'utente (in conformità all'insieme V). Vengono successivamente definiti, sempre dall'utente, due parametri ($minSize$ e $maxSize$) che definiscono la minima e la massima taglia dei moduli in T, cioè il numero di nodi in un δ_p.

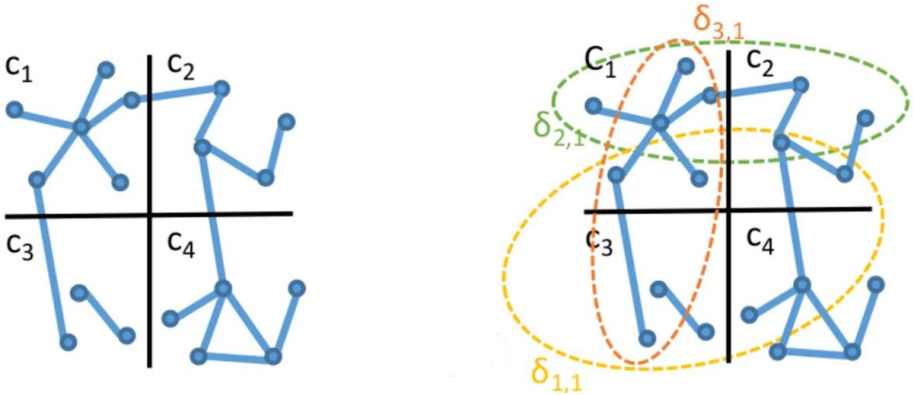


***Figura 7.5**: Le fasi di MTGO.*

Ad ogni **iterazione** viene calcolata una coppia (C, Φ), in particolare C ri-assegnando i nodi alla partizione precedente e Φ selezionando elementi da T che descrivono al meglio C. Ogni partizione di

C è costituita da c_h moduli topologici con h rappresentante l'indice di una singolo modulo topologico ($1 \leq h \leq H$, il numero totale dei moduli funzionali H varia ad ogni iterazione). Idealmente, MTGO tende ad assegnare i nodi in modo che i moduli topologici coincidano con quelli funzionali[19]).

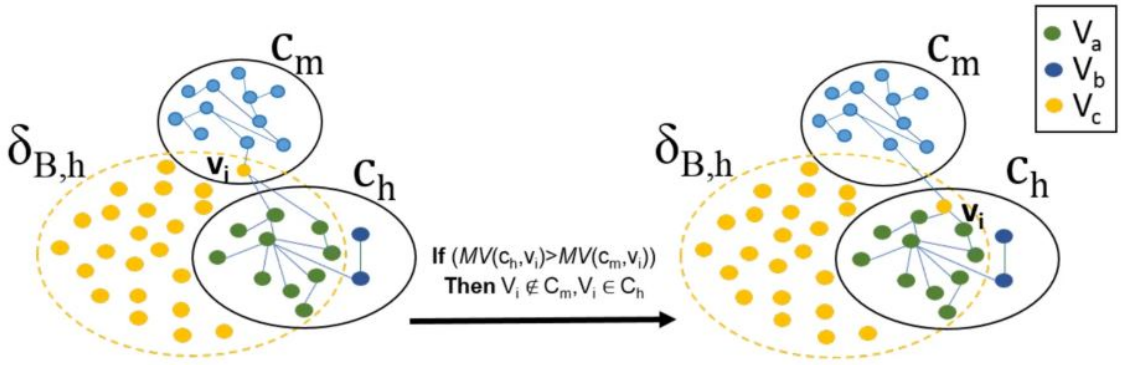


***Figura 7.6**: Riassegnamento di un nodo.*

Per valutare se la **convergenza** è stata raggiunta o meno si utilizzano due funzioni: *modularità* (Q) e *Quality GO* (QGO): Q valuta la qualità globale della partizione C, QGO valuta invece la corrispondenza e il conseguente *overlapping* tra C e Φ. Idealmente, C e Φ dovrebbero essere soggette ad *overlapping*.

La formula di Q è:

$$Q(C^k) = \sum_{1<h<H_k} \frac{e_h^k}{|E|} - \left(\frac{d_h^k}{2 \times |E|}\right)^2.$$

Questa formula serve a valutare le partizioni dei grafi. L'indice k indica la k-esima iterazione dell'algoritmo. C^k è la k-esima partizione, H^k è il numero di moduli topologici, e_h^k è il numero totale di archi nell'h-esimo modulo topologico mentre d_h^k è la somma dei gradi dei nodi dell'h-esimo modulo topologico.

Il valore di Q varia da -1 a 1, i valori positivi (negativi) indicano un maggior (minor) numero di collegamenti all'interno dei moduli topologici rispetto ad una randomizzazione.

MTGO rimane, a quasi 3 anni dallo sviluppo, sconosciuto ai più con solamente 13 citazioni registrate in **Google Scholar**. Dato comunque l'esiguo numero di citazioni, questo metodo rimane il metro di paragone utilizzato da altri ricercatori nello sviluppo di efficienti algoritmi per l'identificazione di moduli funzionali nelle PPIN.

[19]Per una trattazione dettagliata di questa fase rimandiamo alla lettura delle due sottofasi descritte nel dettaglio nel paper di riferimento (si veda la **Bibliografia**).

8 IsoRank

IsoRank è un metodo per l'allineamento globale di più PPIN. L'intuizione da tener presente è che una proteina in una rete PPI produce una buona corrispondenza (*match*) con una proteina in un'altra rete se le loro rispettive sequenze e i loro intorni topologici costituiscono una buona corrispondenza. IsoRank è stato utilizzato per calcolare un allineamento globale delle reti PPI *Saccharomyces cerevisiae* (lievito di birra -fungo-), *Drosophila melanogaster* (moscerino della frutta), *Caenorhabditis elegans* (verme), *Mus musculus* (topo comune) e *Homo sapiens*.

I sottografi individuati con questi allineamenti sono più grandi e più vari di quelli prodotti da metodi precedenti che hanno dimostrato la loro efficacia nell'identificare pattern localizzati confrontando tra di loro solamente due reti. Questo metodo rappresenta un grande passo avanti per l'allineamento di più reti PPI ed è applicabile in molti settori scientifici.

IsoRank rappresenta un approccio di analisi comparativa delle reti PPI al fine di trovare una soluzione al problema di allineamento ottimo *globale* tra due o più PPIN, mirando ad identificare la corrispondenza tra i nodi e gli archi delle reti in input che massimizzi il *match* totale tra le reti.

8.1 *Global* vs. *Local Network Alignment*

In generale, l'obiettivo in un problema di allineamento di rete è quello di trovare un sottografo comune (cioè un insieme di archi "conservati") tra le reti in input. Corrispondentemente a questi archi conservati, esiste una mappatura (*mapping*) tra i nodi delle reti. Per esempio, quando la proteina a_1 dalla rete G_1 viene mappata sulle proteine a_2 in G_2 e a_3 in G_3, allora a_1, a_2 e a_3 si riferiscono allo stesso nodo nell'insieme degli archi conservati. Ciò che rende difficile il problema è il compromesso (*trade-off*) da ottenere: massimizzare la sovrapposizione (*overlap*) tra le reti (cioè il numero di archi conservati), garantendo al tempo stesso che le proteine mappate siano il più possibile correlate.

Mentre un algoritmo *Local Network Alignment* è essenzialmente destinato a trovare pattern simili tra due reti (o sottoreti), l'obiettivo nel *Global Network Alignment* è quello di trovare il miglior allineamento complessivo tra le reti in ingresso. Inoltre, deve valere una proprietà di *transitività*: se a_1 in G_1 viene mappata su a_2 in G_2 e a_2 viene mappata su a_3 e a_4 in G_3, allora anche a_1 dovrebbe essere mappata su a_3 e a_4. Il GNA può essere usato per confrontare gli interattomi e per comprendere le variazioni tra specie.

8.2 *Score* e *mapping*

Consideriamo un semplice caso di GNA a coppie. L'input consiste in due PPIN G_1 e G_2 (ogni arco e può aver associato un peso $w(e)$, con $0 \leq w(e) \leq 1$) e di una *similarity measure* tra i nodi delle due reti (per esempio BLAST *similarity measure*).

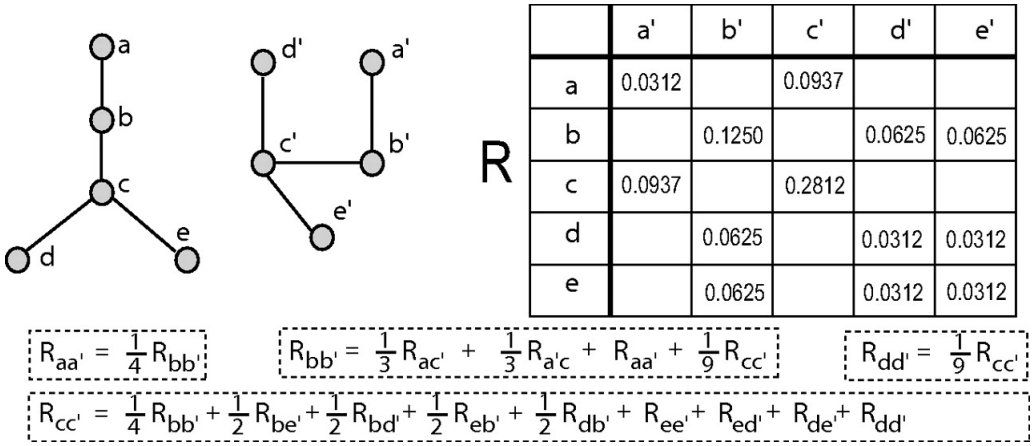
L'output desiderato è un mapping tra i nodi delle due reti che massimizza la combinazione convessa delle seguenti funzioni obiettivo: (1) la dimensione del grafo in comune in seguito al mapping e (2) la somiglianza tra le sequenze dei nodi mappati gli uni negli altri.

L'algoritmo prevede **due fasi**. Nella prima fase associa un *functional similarity score* ad ogni possibile match tra i nodi delle due reti. Sia R_{ij} lo *score* per la coppia di proteine (i, j) dove i proviene dalla rete G_1, mentre j da G_2. La seconda fase costruisce la mappatura per il GNA estraendo un insieme di *score* elevati (in accordo con $\boldsymbol{R}$, il vettore di tutti R_{ij}). Per calcolare il *functional similarity score* R_{ij} consideriamo la coppia (i, j) un "buon match" se le sequenze di i e di j sono allineate e i loro "vicini" costituiscono a loro volta un buon match gli uni con gli altri. Bisogna quindi generare un insieme di vincoli e calcolare i *neighborhood scores* in modo ricorsivo. Si consideri la seguente equazione:

$$\boldsymbol{R} = \sum R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(i)||N(j)|} R_{ij} \text{ con } i \in V_1,\ j \in V_2$$

dove $N(a)$ rappresenta tutti i *neighbors* del nodo a; $|N(a)|$ la cardinalità di questo insieme; V_1 e V_2 sono gli insiemi dei nodi nelle reti G_1 e G_2 rispettivamente. Lo *score* R_{ij} dipende dagli *score* dei vicini di i e j, che a loro volta dipendono dai vicini dei vicini etc.

I nodi che hanno una buona corrispondenza hanno *score* R_{ij} più alti.



	a'	b'	c'	d'	e'
a	0.0312		0.0937		
b		0.1250		0.0625	0.0625
c	0.0937		0.2812		
d		0.0625		0.0312	0.0312
e		0.0625		0.0312	0.0312

Figura 8.1*: Intuizione alla base dell'algoritmo: consideriamo una coppia di grafi isomorfi di piccole dimensioni; per ogni possibile accoppiamento (i, j) tra i nodi dei due grafi calcoliamo i valori R_{ij}. Questi valori sono vincolati (constrained) a dipendere dai valori dei "vicini" (si veda l'equazione).*

A questo punto dell'algoritmo abbiamo uno score R_{ij} per ogni coppia di nodi che non sono nella stessa rete; in genere, per il 99% delle coppie-nodo, questo valore è zero. Dopo aver identificato gli score più alti bisogna assicurarsi che il *mapping* mantenga la proprietà di transitività (descritta in precedenza). Il *mapping* si può ottenere in due modi:

1. **One-to-one Mapping**: ogni nodo viene mappato in al massimo un altro nodo (per specie);
2. **Many-to-many**: un nodo può essere mappato in più di un nodo in un'altra specie.[20]

Concludiamo proponendo l'analisi del sottografo comune ottenuto dall'allineamento delle cinque specie elencate in precedenza.

Il sottografo corrispondente all'allineamento globale possiede 1663 archi in comune ad almeno due PPIN e 157 archi in comune al almeno 3 PPIN. La dimensione del sottografo comune è relativamente piccola (*overlap* solamente con $\approx 5\%$ della PPIN umana) a causa delle probabili incompletezza e rumorosità dei dati. All'aumentare della quantità e della qualità dei dati, l'*overlap* dovrebbe aumentare sensibilmente. Delle 86932 proteine provenienti dalle 5 specie, 59539 (68,5%) hanno ottenuto almeno un match con un'altra proteina di una rete diversa.

IsoRank è stato citato ben 505 volte (fonte: **Google Scholar**) dallo sviluppo nel 2008; proposto in moltissime varianti, costituisce un "baluardo" per il GNA.

[20] **Multiple GNA**: Quando l'input consiste di più di due reti, si ripete il processo appena descritto per tutte le possibili coppie di reti fornite in input e successivamente si calcolano i *functional similarity scores* $\boldsymbol{R}$ per ogni coppia di reti in input.

9 L-GRAAL

Sviluppato nel 2015, L-GRAAL -*Lagrangian graphlet-based network aligner* - è un metodo basato sull'idea di mappare insieme nodi che costituiscono un pattern (presenza di sottografi chiamati *graphlet*[21]) definito da una grande quantità di interazioni condivise. L-GRAAL ottimizza una funzione obiettivo, che fonde le informazioni derivanti dalle sequenze di proteine con le interazioni tra i vari *graphlet*, risolta con la Programmazione Intera. L-GRAAL è in grado di individuare l'*overlap* tra le reti e fornisce risultati migliori di tutti gli altri metodi GO-based a livello di *mapping* delle proteine e delle interazioni tra le stesse.

Date due PPIN, $N_1 = (V_1, E_1)$ e $N_2 = (V_2, E_2)$ (con $|V_1| \leq |V_2|$), un *allineamento globale*, $f : V_1 \to V_2$, è un mapping 1-a-1 dei nodi di V_1 con quelli di V_2. Ad un *allineamento globale* viene associato uno score S:

$$S(f) = \sum_{u \in V_1} n(u, f(u)) + \sum_{(u,v) \in E_1} e(u, f(u), v, f(v))$$

dove $n : V_1 \times V_2 \to \mathbb{R}^+$ corrisponde allo score del mapping tra un nodo di V_1 e uno di V_2, mentre $e : E_1 \times E_2 \to \mathbb{R}^+$ corrisponde allo score del mapping tra un arco di E_1 e uno di E_2. Il *Global Network Alignment problem* cerca di trovare un allineamento globale in grado di massimizzare S.

9.1 *Similarity scores* e funzione obiettivo

In L-GRAAL si misura la relazione tra due proteine mappate u e $f(u)$ utilizzando il loro allineamento di sequenza BLAST[22]:

$$n(u, f(u)) = \frac{seqsim(u, f(u))}{max_{i,j} seqsim(i, j)}$$

dove *seqsim* può essere un qualsiasi tipo di *sequence-based similarity score*. Viene successivamente misurata la *topological similarity* fra due proteine u e $f(u)$ utilizzando il loro da 2- a 4-*node graphlet degree similarity*[23] t:

$$t(u, f(u)) = \frac{1}{15} \sum_{i=0}^{14} \frac{min(d_u^i, d_{f(u)}^i)}{max(d_u^i, d_{f(u)}^i}$$

Viene calcolata la *topological similarity* fra due interazioni mappate (archi), (u, v) e $(f(u), f(v))$, in base al loro *graphlet degree similarity* dei loro nodi terminali mappati:

$$e(u, f(u), v, f(v)) = \frac{1}{2}(t(u, f(u)) + t(v, f(v)))$$

Lo score $e(u, f(u), v, f(v)) \in [0, 1]$.

[21] I *graphlet* sono sottografi di piccole dimensioni, connessi, non-isomorfici, indotti da un grafo più grande (sono indicati come G_0,...,G_{29} nella ***Figura 10.1***).

[22] BLAST è un algoritmo usato per comparare le informazioni contenute nelle strutture biologiche primarie.

[23] I *graphlet* generalizzano il concetto di "grado di un nodo": il *graphlet degree* di un nodo v, indicato con d_v^i è il numero di volte che il nodo v tocca un *graphlet* all'orbita i. Vengono utilizzati per misurare la distanza tra due reti e le *topological similarities* tra i nodi nelle reti.

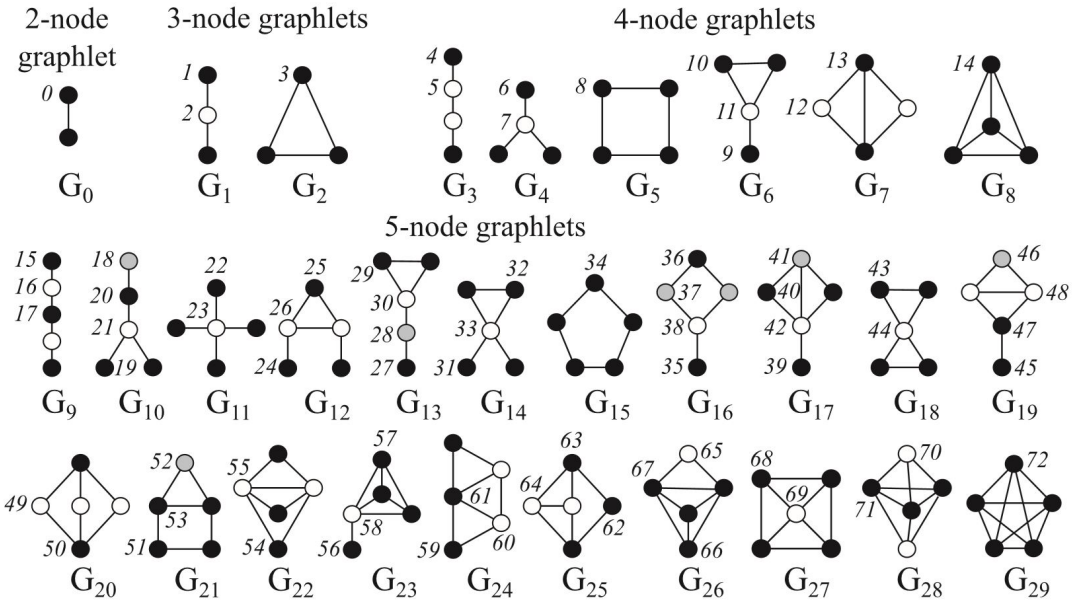


***Figura 10.1**: 2- to 5- graphlets e le loro automorphism orbits.*

La funzione obiettivo di L-GRAAL, S, favorisce la relazione tra le proteine che vengono mappate o tra la *topological similarity* tra le interazioni mappate, in accordo ad un parametro di *balancing* $\alpha \in [0, 1]$:

$$S(f) = \alpha \times \sum_{u} n(u, f(u)) + (1 - \alpha) \times \sum_{u,v} e(u, f(u), v, f(v))$$

9.2 Strategia di ricerca *two-step alignment*

A causa della grande dimensione delle PPIN, risolvere il problema dell'allineamento delle reti cercando di considerare tutti i possibili mapping tra i nodi è, dal punto di vista computazionale, impossibile. L-GRAAL utilizza *sequence* e *graphlet degree similarities* per selezionare un sottoinsieme di nodi mappati $u \longleftrightarrow v$ di modo che $\alpha n(u, v) + (1 - \alpha)t(u, v) \geq 0.5$. L'algoritmo applica successivamente un algoritmo *greedy* per estendere la ricerca degli allineamenti. Il *network alignment problem* può essere espresso tramite la seguente formula:

$$\text{IP} = max_{x,y}\Big(\alpha \sum n(i, k) \times x_{ik} + (1 - \alpha) \sum e(i, j, k, l) \times y_{ijkl}\Big)$$

soggetta ai vincoli:

$$\sum_{k \in V_2} x_{ik} \leq 1, \forall i \in V_1,$$
$$\sum_{i \in V_1} x_{ik} \leq 1, \forall k \in V_2,$$
$$x_{ij} - y_{ijkl} \geq 0, \forall (i, j) \in E_1, \forall (k, l) \in E_2,$$
$$x_{ik} - y_{ijkl} \geq 0, \forall (i, j) \in E_1, \forall (k, l) \in E_2$$

dove, al *node mapping* selezionato, $i \longleftrightarrow k$ con $i \in V_1$, $k \in V_2$, viene associata una variabile binaria $x_{ik} = 1$ se il *node mapping* appartiene all'allineamento, 0 altrimenti. In modo del tutto simile, associamo ad ogni *edge mapping* selezionato $(i, j) \longleftrightarrow (k, l)$, $(i, j) \in E_1$, $(k, l) \in E_2$, una variabile binaria $y_{ijkl} = 1$ se l'*edge mapping* appartiene all'allineamento, 0 altrimenti. I primi due

vincoli obbligano un nodo in V_1 ad essere mappato al massimo in un nodo di V_2 e viceversa, mentre gli ultimi due vincoli obbligano l'*edge mapping* $(i, j) \longleftrightarrow (k, l)$ ad avere gli *end-nodes* mappati: $i \longleftrightarrow k$ e $j \longleftrightarrow l$.

Si risolve l'equazione

$$LR(\lambda) = max_{x,y} \sum n^{\lambda}(i,k) \times x_{ik} + \sum e^{\lambda}(i,j,k,l) \times y_{ijkl}$$

con la Programmazione Intera in $O(|V|^3 + |V|^2 \times d^3)$, dove $|V|$ è il numero di nodi nelle reti e d è il valore del grado (di un nodo) massimo. Risolvere $LR(\lambda)$ genera una soluzione rilassata del problema, che corrisponde ad un *upper bound* alla Programmazione Intera.

Per risolvere tale problema nella sua interezza è necessario ricondursi alla formulazione duale per minimizzazare $LR(\lambda)$ su λ) con la tecnica del *gradient descent*. Sfortunatamente, anche questo problema è $\mathcal{NP}$-completo e, in pratica, si risolve generando una sequenza finita di interazioni. L'algoritmo restituisce il risultato dopo un determinato limite temporale o dopo un numero di iterazioni fissato.

Tutti i test, eseguiti su di un computer desktop con processore Intel Core i7-2600 a 3.40 GHz e una memoria RAM da 64 GB, hanno mostrato il grande potenziale di L-GRAAL (come mostrato dalla ***Figura 10.2***).

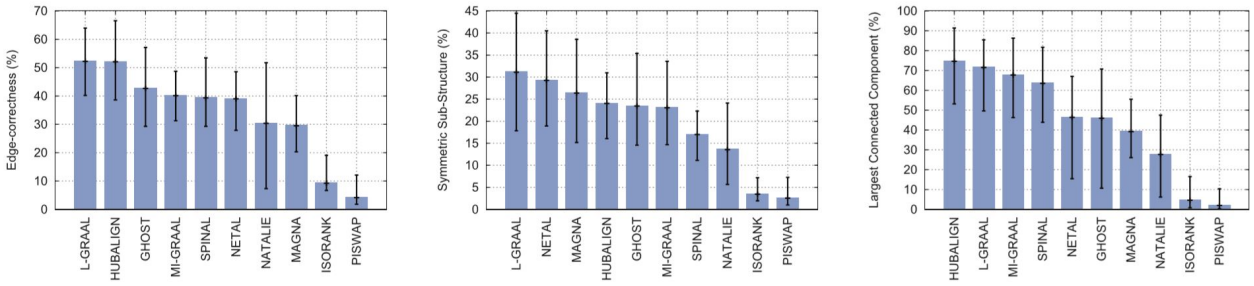


***Figura 10.2**: Confronto L-GRAAL vs altri metodi.*

I metodi considerati sono stati inizialmente confrontati in base alla loro capacità di mappare le proteine che sono collegate in modo simile in entrambe le reti PPI. Nel primo grafico viene mostrata la percentuale di interazioni dalla rete più piccola mappate ad interazioni nell'altra rete. Poiché questa percentuale può essere ottenuta mappando regioni sparse della rete più piccola in regioni densamente connesse di quella più grande, misuriamo, contemporaneamente, anche quanto sono topologicamente simili le regioni mappate usando *symmetric sub-structure score*, che è la percentuale degli archi conservati tra la rete più piccola e la sottorete dalla rete più grande indotta dall'allineamento (si veda il secondo grafico). Infine, utilizziamo la dimensione del componente connesso più grande (*largest connected component*, LCC) per garantire che gli allineamenti corrispondano ad una grande sottostruttura, comune, connessa, invece che a diverse piccole sottostrutture disconnesse.

In tutti i test svolti L-GRAAL (citato 95 volte in **Google Scholar** negli ultimi anni 5 anni) ha mostrato una percentuale di successo non indifferente ed uguale (a volte addirittura superiore) a tutti gli altri metodi con cui è stato confrontato (si pensi ad esempio ad IsoRank, presentato nella ***Sezione 8***).

10 *struc2vec*

La *structural identity* (traducibile con *identità strutturale*) corrisponde ad un concetto di simmetria nel quale i nodi di una rete vengono identificati in base alla struttura della rete stessa e tramite relazioni con altri nodi.

struc2vec è un framework flessibile per l'apprendimento di *latent representations* (= tutte le informazioni importanti necessarie per rappresentare i dati originali) per l'identità strutturale dei nodi[24]. *struc2vec* utilizza una gerarchia, definita dalla sequenza ordinata dei gradi dei nodi, per misurare la *similarity* dei nodi stessi e costruisce un grafo multi-livello (*multilayer graph*) per codificare le somiglianze strutturali.

Sviluppato nel 2017, *struc2vec* presenta prestazioni molto elevate nell'acquisizione di nozioni di identità strutturale in quanto supera i limiti raggiunti dagli approcci precedenti. Gli esperimenti numerici indicano che *struc2vec* migliora le prestazioni su attività di classificazione che dipendono principalmente dall'identità strutturale; *struc2vec* eccelle anche quando la rete originale è soggetta a forti rumori casuali (e.g. rimozione casuale di archi dal grafo).

In quasi tutte le reti, i nodi tendono ad avere una o più funzioni che determinano il loro ruolo nel sistema; come abbiamo imparato, le proteine in una rete di interazione proteina-proteina (PPIN) esercitano funzioni specifiche. Intuitivamente, dunque, diversi nodi in tali reti possono eseguire funzioni simili e spesso possono essere partizionati in classi equivalenti rispetto alla loro funzione nella rete.

***Figura 9.1**: Esempio di due nodi (u e v) strutturalmente simili (gradi 5 e 4, connessi a 3 e 2 triangoli, collegati al resto della rete da due nodi), ma molto distanti nella rete.*

10.1 *struc2vec*: idee di base

Elenchiamo qui di seguito le idee fondamentali che stanno alla base di questo metodo:

1. Valuta la *structural similarity* tra i nodi indipendentemente da eventuali attributi associati a nodi o archi, o addirittura dalla loro posizione all'interno della rete. Due nodi che hanno una struttura locale simile saranno considerati "strutturalmente simili", indipendentemente dalla posizione di rete e dai *labels* associati ai vari *neighborhoods*.
   Questo approccio inoltre non richiede un grafo connesso, identifica nodi strutturalmente simili anche in componenti connesse diverse.
2. Stabilisce una gerarchia per valutare la *structural similarity*. Ai livelli inferiori della gerarchia la *structural similarity* tra i nodi dipende solamente dai loro gradi, per poi, man mano che si procede verso la cima, dipendere dall'intera rete (dal punto di vista del nodo).

[24] Nodi che hanno *neighborhoods* con insiemi simili di nodi dovrebbero avere *latent representations* simili. Ma il *neighborhood* è un concetto locale definito da una nozione di "prossimità" nella rete. Per questo motivo, due nodi "vicini" che sono strutturalmente simili ma molto distanti nella rete avranno *latent representations* diverse. Si veda la ***Figura 9.1***.

3. Genera sequenze di nodi strutturalmente simili (in seguito ad una *random walk* pesata che attraversa un grafo multilayer (e non la rete originale). Pertanto, due nodi che appaiono frequentemente con sequenze simili avranno molto probabilmente una struttura simile.

Consideriamo ora il problema delle *learning representations* che catturano l'identità strutturale dei nodi nella rete. Un approccio corretto dovrebbe presentare queste due proprietà:

- La distanza tra la *latent representation* dei nodi dovrebbe essere fortemente correlata alla loro somiglianza strutturale. Per questo motivo, due nodi con strutture di rete locale identiche dovrebbero anche avere la stessa *latent representation*, mentre i nodi con identità strutturali diverse dovrebbero essere lontani fra loro.
- La *latent representation* non deve dipendere da alcun attributo di nodi o archi, inclusi i *labels* dei nodi. Dunque, nodi strutturalmente simili devono avere una rappresentazione latente stretta, indipendente dagli attributi del nodo e degli archi nel proprio *neighborhood*. L'identità strutturale dei nodi deve essere indipendente dalla sua "posizione" nella rete.

Tenendo presente queste due proprietà, *struct2vec* può essere suddiviso in quattro step principali:

1. **Determinazione della somiglianza strutturale** tra ogni coppia di vertici nel grafo per dimensioni di *neighborhood* diverse; questo fatto porta alla generazione di una gerarchia che fornisce maggiori informazioni per valutare la somiglianza strutturale ad ogni livello della stessa.
2. **Costruzione di un grafo multi-livello pesato** in cui tutti i nodi della rete compaiono in ogni layer e ogni layer corrisponde ad un livello della gerarchia nella misurazione della *structural similarity*.
3. **Utilizzo del grafo multi-livello** per generare (tramite *random walks*) sequenze di nodi. È probabile che queste sequenze includano nodi che sono più strutturalmente simili di altri.
4. Data la sequenza di nodi, **apprendimento della *latent representation***.

10.2 *struc2vec*: descrizione delle iterazioni

Il primo passo compiuto dall'algoritmo consiste nel determinare l'identità strutturale tra due nodi senza utilizzare attribuiti di nodi o di archi. Intuitivamente, due nodi che hanno lo stesso grado sono strutturalmente simili, ma se i loro vicini hanno anch'essi lo stesso grado, allora sono ancora di più strutturalmente simili.

Sia $G = (V, E)$ un grafo non orientato e non pesato dove V è l'insieme dei vertici ed E quello degli archi, dove $n = |V|$ indica in numero dei nodi e k^* il diametro (= la più grande distanza tra coppie di nodi del grafo). Sia $R_k(u)$ l'insieme dei nodi a distanza esattamente $k \geq 0$ da u in G. $R_1(u)$ denota l'insieme dei vicini di u e in generale $R_k(u)$ denota l'anello di nodi distanti k. Sia $s(S)$ la sequenza ordinata di gradi di un *set* $S \subset V$ di nodi. Comparando le sequenze di gradi ordinate degli anelli a distanza k fra u e v possiamo indicare con $f_k(u, v)$ la *structural distance* tra u e v considerando i loro k-hop *neighborhoods* (tutti i nodi a distanza minore o uguale a k e tutti gli archi tra di loro). In particolare, definiamo:

$$f_k(u, v) = f_{k-1}(u, v) + g(s(R_k(u)), s(R_k(v))),$$

$$k \geq 0 \text{ e } |R_k(u)|, |R_k(v)| > 0$$

dove $g(D_1, D_2) \geq 0$ misura la distanza tra le sequenze ordinate D_1 e D_2.

Per confrontare i gradi di due sequenze si utilizza *Dynamic Time Warping*[25]. DTW trova l'allineamento ottimo fra due sequenze A e B. Data una funzione distanza $d(a, b)$ per gli elementi della sequenza, DTW confronta ogni elemento $a \in A$ e $b \in B$, di modo che la somma delle distanze tra elementi corrispondenti (*matched*) sia minima. Dal momento che gli elementi delle sequenze A e B sono i gradi dei nodi, viene adottata la seguente formula:

$$d(a, b) = \frac{max(a, b)}{min(a, b)} - 1$$

Notare che quando $a = b$ allora $d(a, b) = 0$. Per questo motivo due sequenze ordinate di gradi, identiche, avranno distanza pari a 0.

A questo punto è necessario costruire un grafo a più strati pesato che codifichi la *structural similarity* tra i nodi.

Sia M questo grafo dove il layer k viene definito utilizzando i k-hop *neighborhoods* dei nodi. Ogni layer $k = 0, ..., k^*$ è formato da un grafo non orientato completo e pesato con V insieme dei nodi e $\binom{n}{2}$ archi. Il peso dell'arco tra due nodi in un layer è dato dalla seguente formula:

$$w_k(u, v) = e^{-f_k(u,v)},\ k = 0, ..., k^*.$$

I nodi che saranno strutturalmente simili ad u avranno pesi maggiori tra i vari layer di M. I layer vengono collegati utilizzando archi orientati: ogni vertice è collegato al corrispondente nei layer superiore ed inferiore. Perciò, ogni vertice $u \in V$ nel layer k è collegato al corrispondente vertice u nei layer $k + 1$ e $k - 1$. Il peso degli archi tra i layer è definito in questo modo:

$$w(u_k, u_{k+1}) = log(\Gamma_k(u) + e),\ k = 0, ..., k^* - 1$$

$$w(u_k, u_{k-1}) = 1,\ k = 1, ..., k^*$$

dove $\Gamma_k(u)$ è il numero di archi incidenti in u che hanno peso maggiore del peso medio di un arco nel grafo completo al layer k. Per ogni nodo u *struc2vec* procede con una *random walk* partendo dal layer 0. Queste *random walks* hanno una lunghezza (= numero di passi) fissata e relativamente corta; il processo viene ripetuto diverse volte, dando luogo a *walks* multiple ed indipendenti.

Si utilizza infine *Skip-Gram*, una tecnica di *unsupervised learning* per NLP, per identificare o derivare le *latent representations* dalle sequenze ottenute.

A *struc2vec* è possibile applicare una serie di ottimizzazioni (a partire dall'implementazione di DTW, complessità di $O(l)$ a fronte dell'$O(l^2)$ iniziale) che permettono di raggiungere una complessità totale di $O(k^* n^3)$ grazie alla riduzione della lunghezza delle sequenze dei gradi e del numero dei layer.

10.3 *struc2vec* vs *DeepWalk* e *node2vec*

struc2vec è stato testato in diversi scenari e confrontato con gli algoritmi allo stato dell'arte (*DeepWalk* e *node2vec*) per l'apprendimento di *latent representations*.

Il primo test ha previsto la costruzione di un particolare tipo di grafo, denominato *barbell graph*, costituito da due grafi completi connessi da un *path graph*. Ogni coppia di nodi che è strutturalmente equivalente dovrebbe avere *latent representations* simili (che a loro volta dovrebbero essere in grado di descrivere, nel miglior modo possibile, la gerarchia strutturale).

[25] Algoritmo che permette l'allineamento tra due sequenze, e che può portare ad una misura di distanza tra le due sequenze allineate.

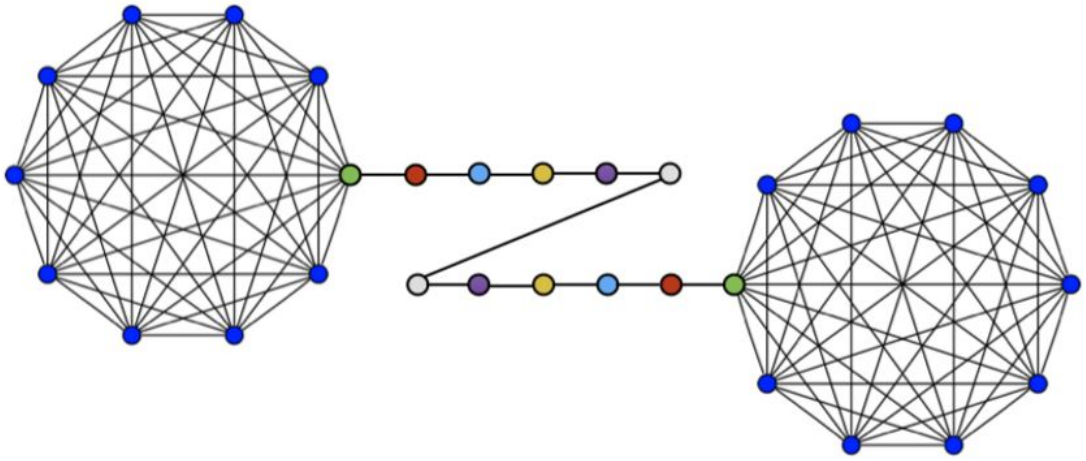


***Figura 9.2**: Barbell graph.*

Anche in seguito ad un tuning dei parametri, *DeepWalk* fallisce nell'individuare le equivalenze strutturali, mentre *node2vec* non riconosce le identità strutturali; *struc2vec* invece individua le *latent representations* posizionando i nodi strutturalmente equivalenti gli uni vicino agli altri (si veda la ***Figura 9.3***).

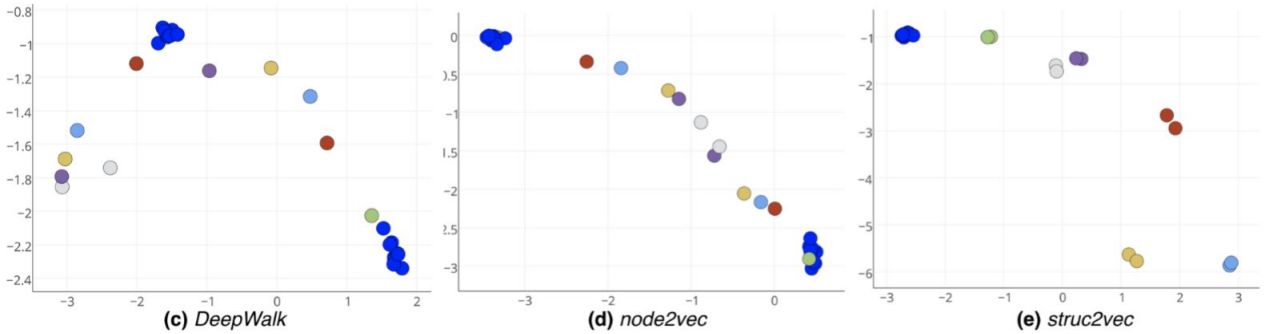


***Figura 9.3**: Confronto di struc2vec con due algoritmi allo stato dell'arte; il metodo eccelle nell'individuazione di structural identity fra nodi.*

Un secondo test ha previsto l'utilizzo della *Zachary's Karate Club network*, una rete composta da 34 nodi e 78 archi, nella quale ogni nodo rappresenta un membro del club e gli archi denotato un'interazione (esterna al club) fra due membri (informalmente, una relazione di "amicizia"). La rete è stata duplicata (si veda la ***Figura 9.4***) in due grafi G_1 e G_2 nei quali ogni nodo in G_1 possiede un corrispettivo *specchio* in G_2. I due grafi sono stati connessi tramite un arco fra i nodi 1 e 37.

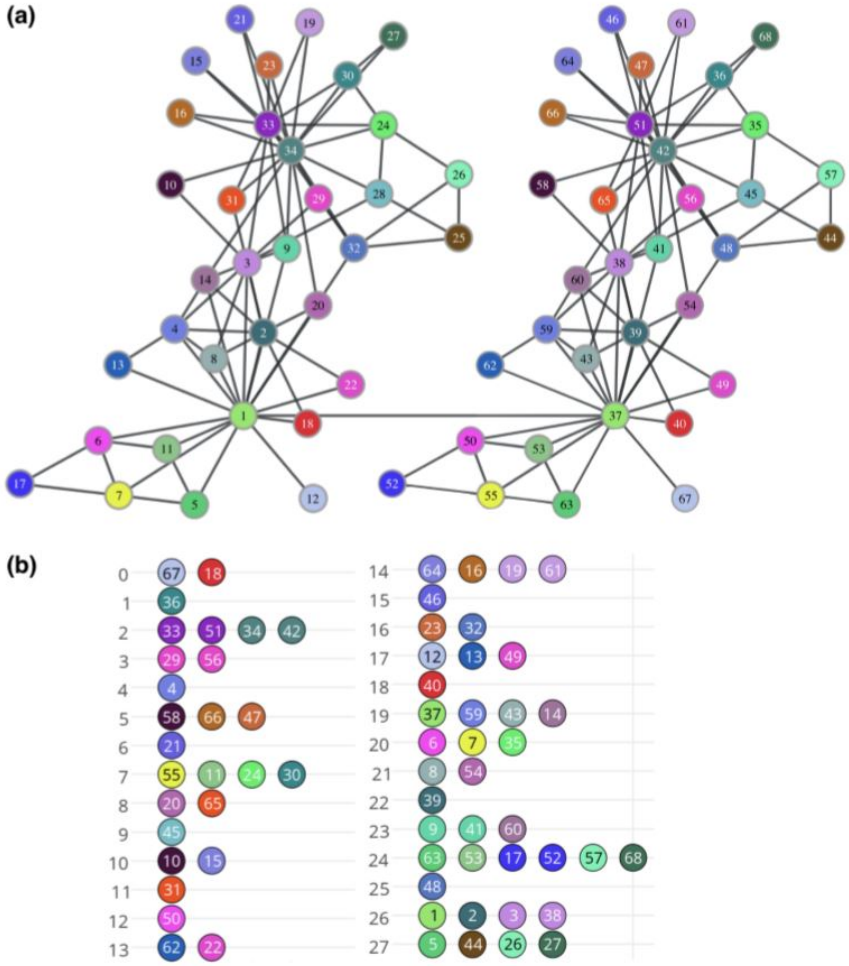


***Figura 9.4**: Zachary's Karate Club network con nodi specchio.*

Anche in questo caso *DeepWalk* e *node2vec* falliscono nell'individuare le *latent representations* di nodi strutturalmente equivalenti (inclusi i nodi *specchio*), mentre *struc2vec* fornisce i risultati migliori.

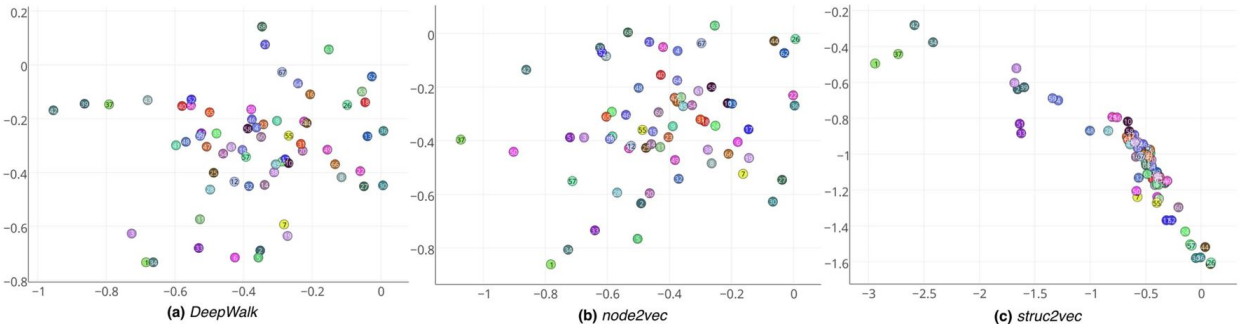


***Figura 9.5**: Secondo test: confronto di struc2vec con i due algoritmi allo stato dell'arte.*

Per concludere, ci teniamo a sottolineare il fatto che *struc2vec* è stato citato 302 volte (fonte: **Google Scholar**) e per gli ambiti più differenti. È stato confrontato con un metodo sviluppato l'anno successivo, *Deep Recursive Network Embedding* (DRNE), sullo stesso dataset; le prestazioni rimangono tutt'ora molto elevate.

11 Conclusioni

Negli ultimi anni, il corpus di dati PPI è cresciuto esponenzialmente e il rapido ritmo di accumulo dati continua imperterrito tutt'oggi. L'obiettivo di questo progetto è stato di far capire la struttura delle reti di interazione proteina-proteina e le implicazioni dal punto di vista biologico.

Scoprire e capire i pattern all'interno delle PPIN è un problema centrale in Biologia. Gli allineamenti tra queste reti permettono di scoprire informazioni su complessi proteici che fino a pochi anni fa non erano note.

Abbiamo proposto una descrizione, seppur breve, del funzionamento di quattro dei metodi più all'avanguardia proposti negli ultimi anni: MTGO, IsoRank, L-GRAAL e *struc2vec*; metodi con un'elevata complessità dal punto di vista computazionale a causa dell'$\mathcal{NP}$-completezza del problema.

Molte sfide sono ancora aperte e molte frontiere devono ancora essere esplorate; con questo progetto abbiamo solamente dato una vaga idea della vastità dell'argomento, di cui si è appena iniziato a parlare.

Luca Masiero
Stefano Ivancich

12 Bibliografia & Sitografia

1. https://www.ebi.ac.uk/training/online/course/network-analysis-protein-interaction-data-introduction
2. https://www.ebi.ac.uk/training/online/course/goa-and-quickgo-quick-tour/what-go
3. Vella, D., Marini, S., Vitali, F. et al. MTGO: PPI Network Analysis Via Topological and Functional Module Identification. Sci Rep 8, 5499 (2018). https://doi.org/10.1038/s41598-018-23672-0
4. Mitra, K., Carvunis, A., Ramesh, S. et al. Integrative approaches for finding modular structure in biological networks. Nat Rev Genet 14, 719–732 (2013). https://doi.org/10.1038/nrg3552
5. Noël Malod-Dognin, Nataša Pržuli. L-GRAAL: Lagrangian graphlet-based network aligner. Department of Computing, Imperial College London, London, UK (2015)
6. Ribeiro, Saverese, Figueiredo. *struc2vec*: Learning Node Representations from Structural Identity. Federal University of Rio de Janeiro, Systems Eng. and Comp. Science Dep. (2017)
7. Singh, Xu, Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. Computer Science and Artificial Intelligence Laboratory and Department of Mathematics, Massachussets Institure of Technology, Cambridge, MA 02139; and Toyota Technological Institute at Chicago, Chicago, IL 60637 (2008)