Hypergraph Extensions for PPI Networks

Valerio Di Pasquale

Abstract

Nel corso degli ultimi anni lo sviluppo del *deep learning geometrico* ha portato ad una importante crescita degli studi riguardo le capacità di queste tipologie di reti neurali nella risoluzione di complessi problemi legati ai grafi e ai sistemi rappresentabili come grafi o in generale non rappresentabili opportunamente con strutture dati "tradizionali". Tuttavia, nonostante l'elevato potenziale di questi modelli rispetto a quelli tradizionali, la capacità di queste particolari reti neurali, così come avviene per le reti tradizionali, è comunque influenzata dalla qualità e dalla struttura dei dati sui quali queste lavorano.

In questo lavoro, si andrà a considerare un possibile metodo per arricchire i dati strutturati come grafi utilizzando pattern strutturali nascosti, al fine di migliorare le performance di diversi task applicati ad essi; in particolare ci focalizzeremo sulle PPI network, i cui nodi rappresentano delle proteine e gli archi descrivono una interazione o una certa soglia di similarità tra le proteine. Lo scopo è enfatizzare l'importanza dei motif, e più nello specifico di questo lavoro, le clique di diversa taglia, come pattern strutturale per le reti biologiche.

Glossario

$\mathcal{H} = (V, E)$
$\mathcal{G} = (V, E)$
V Insieme dei nodi di un grafo o di un ipergrafo
E Insieme di archi o di iperarchi
A Matrice di adiacenza di un grafo G
H Matrice di incidenza di un ipergrafo H
$\mathcal{N}(v)$ insieme dei vicini di un nodo v.
$deg(v)/deg(e)$ grado di un nodo v/iperarco e.

1. Introduzione

I grafi, permettono di rappresentare la complessa organizzazione dei sistemi biologici sotto forma di reti in cui vigono relazioni binarie tra le biomolecole. Ad esempio, nelle

PPI network (protein-protein interaction networks), coppie di proteine, rappresentate dai nodi del grafo, sono collegate da archi che descrivono interazioni tra esse Zitnik et al., 2024. Le *PPI network* possono essere utilizzate per rappresentare complessi proteici di ampia scala in cui vengono definiti legami tra le proteine tramite le loro relazioni fisiche o funzionali (Grindrod & Kibble, 2004).

Queste reti, sono molto utili, in quanto consentono di strutturare questi dati e migliorare le applicazioni biologiche e biomediche. Si basano sul ruolo delle proteine nelle funzioni biologiche, le loro interazioni determinano meccanismi molecolari e cellulari che controllano lo stato di salute degli organismi. Pertanto, tali reti facilitano la comprensione dei meccanismi patogeni (e fisiologici) che innescano l'insorgenza e la progressione delle malattie. Di conseguenza, possiamo immaginare di applicare le tecniche di studio dei grafi per estrapolare conoscenza che può essere tradotta in efficaci strategie diagnostiche e terapeutiche.

Tuttavia, spesso le interazioni binarie che sono definite in queste reti, non sono sufficienti per rappresentare i complessi meccanismi di questi sistemi, dunque si ricade nell'utilizzare sovrastrutture (e.g. multipli livelli di interazione), che colmano il limite espressivo della rappresentazione tradizionale (Battiston et al., 2020). Ciò suggerisce il bisogno di catturare interazioni di alto ordine tra due o più nodi, piuttosto che semplici relazioni binarie così come avviene nei grafi.

Negli ultimi anni, gli ipergrafi, hanno guadagnato una crescente attenzione grazie alla loro flessibilità nel modellare in maniera diretta complessi sistemi in cui sono necessarie relazioni di alto ordine per modellare le interazioni che avvengono tra le entità (Antelmi et al., 2023). In questo lavoro si andranno ad analizzare le performance dei modelli basati su *deep-learning* applicabili per la convoluzione su grafi e ipergrafi, andando a sfruttare le clique per costruire collegamenti artificiali tra i nodi delle reti al fine di valutare quali benefici possiamo trarre da questo tipo di procedura di *data preparation*.

2. Background e notazioni

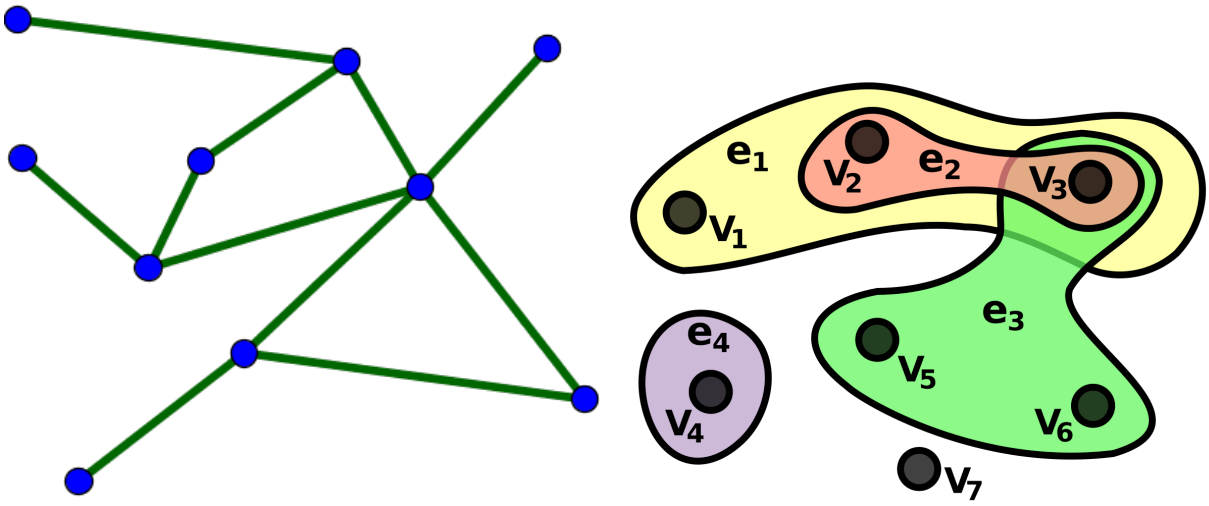
In questa sezione verranno introdotti i concetti base e le notazioni utilizzate per lo studio dei grafi e degli ipergrafi, fondamentali per l'analisi di queste strutture. Verranno esplorati i concetti di motif e graphlet, che consentono di identificare pattern strutturali nascosti all'interno dei grafi. Successivamente, si discuterà del *Geometric Deep Learning*, un approccio avanzato che estende le tecniche di *deep learning* alle strutture dati non euclidee. Infine, verranno esaminati i benefici e le applicazioni di queste tecniche ai grafi che rappresentano sistemi biologici, con particolare attenzione alle *PPI networks*.

2.1 Grafi e Ipergrafi

I grafi $\mathcal{G} = (V, E)$ sono strutture matematiche discrete, spesso rappresentati come una coppia ordinata di due insiemi V ed $E \subseteq V^2$. Dunque, ogni arco $e = (u, v) \in E$ rapp-

resenta una coppia che descrive una relazione binaria tra due nodi $u, v \in V$, ordinata o meno se stiamo considerando un grafo diretto o indiretto.

Gli ipergrafi, così come i grafi, sono strutture matematiche discrete, che in maniera analoga possono essere indicati come una coppia ordinata $\mathcal{H} = (V, E)$, dove V è l'insieme dei nodi e a differenza dei grafi, $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ rappresenta l'insieme degli *iperarchi*, tale che ogni iperarco $e \in E$ è un sottoinsieme non vuoto dell'insieme dei nodi, $e \subseteq V \wedge e \neq \emptyset$.



(a) (b)

Figure 1: La figura a sinistra rappresenta un semplice grafo indiretto (a), nel quale ogni arco, denotato da una linea collega solo due vertici; quella a destra (b), mostra un ipergrafo indiretto, per il quale è possibile osservare come numeri arbitrari di nodi sono messi in relazione da iperarchi.

2.2 Motif e Graphlet

Nel contesto dell'analisi dei grafi, i motif e i graphlet sono strutture di fondamentale importanza, che consentono di comprendere meglio le proprietà strutturali e le dinamiche delle reti.

I **motif** sono sottografi ricorrenti all'interno di un grafo più grande e sono considerati come le unità di base che costituiscono la struttura complessiva della rete (*building-blocks* dei grafi). Questi piccoli schemi sono di particolare interesse in diversi campi, come la biologia, dove possono rappresentare moduli funzionali nelle reti di interazione proteica o circuiti di regolazione genica. L'identificazione e l'analisi dei motif possono rivelare informazioni critiche sul funzionamento e sull'evoluzione delle reti biologiche, oltre a fornire indizi sui meccanismi alla base dei processi biologici. I motif sono statisticamente significativi, ciò significa che il numero delle loro occorrenze in una rete del mondo reale è differente in frequenza rispetto ad un *null-model*.

La significatività di apparizione di un insieme di motif in un grafo G rispetto ad un grafo casuale G_R può essere misurata utiizzando il z-score $Z(\cdot)$.

$$Z(G) = \frac{C_i(G) - \overline{C_i(G_R)}}{\sigma C_i(G_R)}$$

dove $C_i(\cdot)$ indica il numero di occorrenze dell'i-esimo motif.

I **graphlet** sono simili ai motif, ma si differenziano principalmente per due motivazioni: essi sono sottografi indotti, questo significa che, fissato un template di graphlet, se due nodi di esso hanno un arco nel grafo di riferimento, questo deve essere presente anche nel template di *graphlet*; non devono essere statisticamente significativi, ovvero, non è necessario che la loro frequenza di apparizione sia significativa rispetto ad un *null-model*.

2.3 Deep Learning Geometrico

Il *Geometric Deep Learning* è un campo emergente che estende i tradizionali approcci di *deep learning* ai dati non euclidei (Bronstein et al., 2021), come griglie, grafi e ipergrafi. Queste tecniche permettono di risolvere complessi task su queste reti, costruendo algoritmi di apprendimento automatico con lo scopo di massimizzare una funzione obiettivo o minimizzare una funzione di costo, ovvero esattamente come avviene per le *deep neural network* tradizionali. Questi approcci consentono di analizzare strutture dati molto più complesse e che possono rappresentare molte più informazioni rispetto ai semplici dati tabulari.

2.3.1 Graph Convolution

L'operazione di convoluzione per grafi *Graph Convolutional Networn* (GCN) Kipf & Welling, 2016a,

$$\hat{\mathbf{X}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{A}^{\top}\mathbf{X}\mathbf{\Theta}$$

dove $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ indica la matrice di adiacena di un grafo G al quale sono aggiunti archi della tipologia $(v, v)\forall v \in V$; D è la matrice diagonale dei gradi dei nodi tale che $D_{vv} = deg(v) = \sum_{u \in V} \hat{A}_{vu}$ e Θ sono i parametri coinvolti durante la fase di addestramento della rete neurale.

2.3.2 Hypergraph Convolution

L'operatore di convoluzione per ipergrafi *Hypergraph Convolutional Network* (HGCN) Bai et al., 2019

$$\hat{\mathbf{X}} = \mathbf{D}^{-1}\mathbf{H}\mathbf{B}^{-1}\mathbf{H}^{\top}\mathbf{X}\mathbf{\Theta}$$

Dove $H \in \{0, 1\}^{|V|\times|E|}$ è la matrice di incidenza di un ipergrafo $\mathcal{H}$, D e B sono rispettivamente le matrici diagonali dei gradi dei nodi e degli iperarchi, tali che $D_{vv} = deg(v)$ e $B_{ee} = deg(e)$, infine, Θ, così come per le GCN sono i parametri coinvolti nell'apprendimento della rete neurale.

Gli autori dell'operatore HGCN, hanno inoltre dimostrato che l'operazione di convoluzione su grafi, risulta essere un caso speciale di convoluzione su grafi, sottolineando dunque l'equivalenza dei due operatori nel lavorare su un grafo.

3. Stato dell'arte

Come descritto nelle sezioni precedenti, solitamente, la rappresentazione sotto forma di grafi per le strutture biologiche, non è sufficiente per l'esecuzione dei task. Questo ha portato alla costruzione di componenti addizionali che si sovrappongono alle interazioni binarie. In letteratura sono state proposte diverse idee che sfruttano relazioni di alto ordine per compensare le limitazioni dei grafi nella rappresentazione delle strutture biologiche, queste possono essere divise in due principali categorie.

3.1 Higher-order dependencies

Le strutture che rientrano in questa categoria, si basano comunque su grafi, tuttavia fanno anche uso di dipendenze di alto ordine tra coppie di nodii (Xu et al., 2016). o tra piccoli sottografi ('Graphlets in Network Science and Computational Biology', 2019). Per quanto riguarda la prima metodologia, basata sulle dipendenze di alto ordine, è stato osservato che quando si rappresentano dati temporali sottoforma di reti, l'utilizzo delle sole dipendenze di primo ordine può produrre risultati imprecisi (Xu et al., 2016). Ciò, può essere causato dal fatto che i dati campionati da sistemi molto complessi, possono raggiungere un ordine delle interazioni anche di 5, producendo una notevole approssimazione nel descrivere questi sistemi con strutture dati troppo semplici.

3.2 Higher-order coordinated patterns

Il secondo approccio è stato proposto per catturare dipendenze di ordine variabile tra coppie di nodi. I sottografi, possono essere visti come strutture che descrivono relazioni di alto ordine tra due o più nodi di un grafo (Battiston et al., 2020). Esempi di sottografo sono: cammini, triangoli aperti e chiusi, clique di cardinalità arbitraria. Inoltre, i sottografi possono essere divisi in due principali categorie: le graphlet, ovvero grafi indotti e motif, ovvero pattern statisticamente significativi rispetto ad un *null model*. Nel contesto di questo lavoro ci concentreremo su questa particolare tecnica, sfruttando le clique come pattern strutturale delle *PPI networks*.

4. Obiettivi

L'ipotesi alla base degli esperimenti considerati, sta nel rappresentare le informazioni strutturali nascoste presenti in tutte le reti del mondo reale attraverso adeguate strutture dati che ci permettano di sfruttare questa conoscenza per svolgere vari task. Andremo

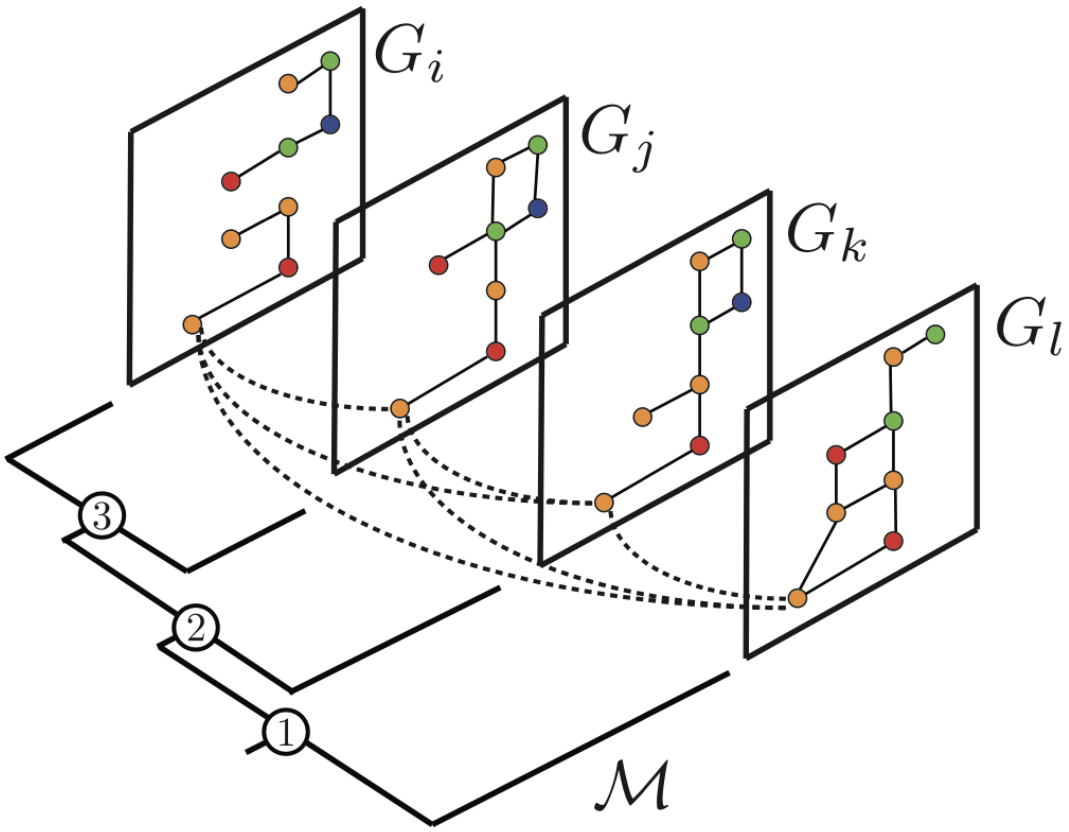


Figure 2: Un grafo *multi-livello* organizzato in quattro livelli

quindi a costruire ipergrafi che permettano di apprendere rappresentazioni dei nodi in relazione ai motif di cui fanno parte.

4.1 Definizione dei problemi

In questa sezione vengono descritti i principali problemi affrontati nel contesto di questo lavoro, fornendo una panoramica delle sfide e delle metodologie adottate per risolverli.

4.1.1 Function Prediction

Il problema di *function prediction* può essere visto come un problema di classificazione multi-etichetta.

Per questo lavoro, è stato considerato il task di *function-prediction* nella versione che tiene conto delle *features* associate ai nodi e nella versione che non ne tiene conto, nella sezione dedicata alla configurazione degli esperimenti 6 è descritto in dettaglio l'approccio utilizzato.

4.1.2 Link Prediction

Il problema di *link prediction*, è uno dei task più frequenti applicati ai grafi di qualsiasi dominio. Eseguire la previsione degli archi significa riconoscere da un insieme di archi

quali abbiano più probabilità di apparire in una specifica rete. Le applicazioni di questo task spaziano dai sistemi di raccomandazione alle reti di natura biologica.

Dato un grafo $\mathcal{G} = (V, E)$, l'obiettivo del task di *link prediction*, è quello di trovare gli archi mancanti che più verosimilmente dovrebbero appartenere ad un insieme E di archi osservati (o noti).

Per un arco e, molti metodi di link prediction mirano a stimare o apprendere una funzione ψ tale che

$$\psi(e) = \begin{cases} \geq \epsilon, \text{if } e \in E \\ < \epsilon, \text{if } e \notin E \end{cases}$$

dove ϵ rappresenta un valore di soglia utilizzato per *binarizzare* il valore continuo di ψ in un etichetta (R. Zhang et al., 2019).

Per i modelli basati su deep learning (GCN e HGCN), il metodo di link prediction utilizzato si basa sulla fattorizzazione della matrice di incidenza A come $\hat{A} = XX^\top$, dove X è una matrice di taglia $|V| \times d$ con d fattori.

5. Configurazione degli esperimenti

In questa sezione viene illustrata la configurazione degli esperimenti condotti, descrivendo dettagliatamente i dataset utilizzati, la costruzione degli ipergrafi a partire dalle reti PPI e gli algoritmi impiegati per la predizione dei link. Vengono presentati i criteri adottati per la trasformazione dei grafi in ipergrafi, e vengono spiegati i diversi indici di similarità utilizzati per la valutazione delle connessioni tra i nodi, tra cui il Coefficiente di Jaccard, l'indice Adamic Adar e il Resource Allocation, motivando come mai la scelta è ricaduta proprio su queste euristiche.

5.1 Dataset considerati

PPI24 Gli autori del dataset (Zitnik & Leskovec, 2017) hanno associato alle proteine dei diversi tessuti basandosi sul lavoro di Greene et al., 2015 sulla Gene Ontology (GO) (Ashburner et al., 2000). Ogni nodo (i.e. le proteine) sono assegnati ad una o più etichette (i.e. *cellular function* estrapolate dalla GO). L'intero dataset è costituito da 24 tessuti (tissues) ognuno rappresentato come un grafo. Ogni nodo della rete descrive una proteina e ad ogni proteina sono associate 50 diverse features oltre che 121 etichette descrittive delle funzioni proteiche.

Come descritto nelle sezioni precedenti, per il task di *function prediction* e di *link prediction* su questo dataset, sono stati considerati due approcci: (1) utilizzando le *feature* associate ai nodi e fornite dal dataset; (2) non utilizzare le *feature* associate ai nodi. Per la metodologia (1), non sono state apportate modifiche al dataset. Invece, per la strategia (2), è stato adoperato un metodo simile a quello

utilizzato dagli autori di (Kipf & Welling, 2016b), utilizzando una matrice di identità **I** come sostituzione della originale matrice delle *features*. Tuttavia, differentemente da questo esatto metodo, che è stato testato su dataset di dimensione relativamente piccola, utilizzarlo nel nostro caso non avrebbe permesso di eseguire correttamente gli esperimenti per via di limitazioni legate all *hardware* delle macchine su cui sono stati eseguiti gli esperimenti. Dunque, l'approccio adoperato consiste nell'assegnare come vettore delle feature un vettore *one-hot* (con un solo termine uguale ad uno e tutti gli altri termini uguali a zero) casuale, della taglia degli originali vettori delle *features* ad ogni nodo.

# Nodes	56944
# Node Features	50
# Node Labels	121
# Edges	1587264

Table 1: Statistiche del dataset PPI24, sono riportati anche il numero di features e di labels che descrivono le funzioni associate ad ogni nodo

Per il task di *function-prediction* dataset è stato diviso in 20 grafi per il training set, 2 per il validation set e 2 per il test set. Per il task di *link-prediction* ogni grafo è stato partizionato con proporzione 70/10/20, rispettivamente per *training*, *validation* e *test set*.

PPI144 Il dataset in questione è molto simile al precedente, l'intero dataset è composto da 147 grafi disgiunti, dove ogni grafo rappresenta un *tissue* che raggruppa proteine umane assegnate alla stessa *tissue* secondo la *Tissue Ontology* (Chang et al., 2014).

Tale dataset è stato utilizzato per testare il task di *link prediction* vista l'assenza di etichette associate ai nodi.

5.2 Costruzione degli Ipergrafi

Al fine di generare degli iperagrafi a partire dai motif delle PPI network dei dataset considerati, vi è la necessità di utilizzare algoritmi che abbiano delle specifiche caratteristiche: (1) apprendere gli score degli iperarchi a partire dall'espazione a clique dell'ipergrafo, questo tipo di vista dell'ipergrafo non è vincolato dal tipo di motif considerato per la costruzione della nuova struttura; (2) non avere la necessità di utilizzare etichette associate agli iperarchi, dunque poter essere addestrato in maniera non-supervisionata; (3) applicabile in maniera diretta/indiretta per la previsione di iperarchi di un ipergrafo.

A tale scopo sono stati selezionati tre popolari algoritmi comunemente utilizzati per la risoluzione del task di *link prediction*. Al fine di utilizzare tali modelli per trasformare un grafo di questo tipo in un ipergrafo, sono stati considerati come possibili iperarchi le clique di ogni ordine formate dai nodi della rete.

Lo score di esistenza delle clique come ipergrafi è stato valutato come la media degli score (per ogni modello di *link prediction* testato) degli archi che formano la clique; a

questo punto per binarizzare la selezione degli iperarchi, è stato calcolato lo score medio di ogni arco della rete, selezionando i soli iperarchi con un valore di score maggiore o uguale ($\geq$) dello score medio dell'intera rete.

	JC	AA	RA
# Hyperedges	349967	364285	388018

Table 2: Statistiche del dataset PPI24 con il numero di iperarchi ottenuto attraverso il processo di costruzione dell

JC il *Coefficiente di Jaccard*, anche noto come coefficiente di similarità di Jaccard (Jaccard, 1901), è un indice statistico utilizzato per confrontare la similarità o la diversità di due insiemi campionari A e B.

$$JC_{uv} = \frac{|A \cap B|}{|A \cup B|}$$

Esso può essere utilizzato come euristica nei grafi per stimare la similarità di due nodi u e v, attraverso il confronto del loro vicinato $\mathcal{N} : V \to \mathcal{P}(V)$.

$$JC_{uv} = \frac{|\mathcal{N}(u) \cap \mathcal{N}(v)|}{|\mathcal{N}(u) \cup \mathcal{N}(v)|}$$

AA *Adamic Adar* è un coefficiente per la previsione di relazioni nelle reti sociali (Adamic & Adar, 2003), esso è definito come la somma della frequenza logaritmica inversa dei vicini comuni tra due nodi di un grafo.

$$AA_{uv} = \sum_{s \in \mathcal{N}(u) \cap \mathcal{N}(v)} \frac{1}{log|\mathcal{N}(s)|}$$

Tale indice sfrutta l'idea che vicini comuni alla maggior parte dei nodi hanno minore significato rispetto ai vicini condivisi da un numero ristretto di nodi.

RA *Resource Allocation* è un indice molto simile al coefficiente *Adamic Adar*, esso è ottenuto come la somma della frequenza inversa dei vicini comuni tra due nodi in un grafo.

$$RA_{uv} = \sum_{s \in \mathcal{N}(u) \cap \mathcal{N}(v)} \frac{1}{|N(s)|}$$

5.3 Addestramento dei modelli

I modelli sono stati addestrati utilizzando l'algoritmo di ottimizzazione *AdamW* con *learning-rate* pari a 0.0025; utilizzando la funzione di costo *Binary Cross Entropy with Logits* come funzione di perdita, tale funzione è simile alla più nota *Binary Cross Entropy*,

con l'unica differenza di combinare in maniera diretta quest'ultima con una funzione sigmoidea, avendo come risultato una discesa del gradiente numericamente più stabile.

Ogni modello è stato addestrato per un massimo di 5000 epoche, utilizzando la tecnica di early stopping basata sulla *pazienza*, applicata alla media mobile della funzione di perdita. Il valore scelto per la pazienza è di 20, la taglia della finestra utilizzata per il calcolo della media mobile è 100. Al termine dell'addestramento vengono utilizzati i pesi del modello che hanno ottenuto il valore minimo per la funzione di perdita.

6. Risultati

Metriche Per i risultati degli esperimenti è stata utilizzata la *ROC* (*Receiver operating caracteristic*), questa metrica viene creatat tracciando il valore del *True Positive Rate* (*TPR*) in relazione a quello del *False Positive Rate* (*FPR*). La scelta di questa metrica è preferibile nei casi in cui si vuole valutare le performance del modello in maniera indipendente rispetto al valore di una *threshold*. In aggiunta, è stato misurato il tempo (in secondi) necessario ad ogni modello per convergere (*E. time (s)*) durante la fase di addestramento.

Function Prediction Gli esperimenti per il task di function prediction, permettono di evidenziare un leggero milgioramento per i modelli che sfruttano le relazioni di alto ordine rispetto al semplice modello che non le utilizza. Il mdello che ottiene performance migliori è HGCN utilizzato sull'ipergrafo generato utilizzando l'indice *Resource Allocation*, il miglioramento ottenuto in questo caso è quasi due punti percentuali superiore rispetto al modello basato su grafi.

	GCN	HGCN (JC)	HGCN (AA)	HGCN (RA)
ROC	94.761 ± 0.68	95.218 ± 1.20	95.981 ± 0.38	**96.113 ± 0.37**
ROC (*)	45.94 ± 0.62	62.52 ± 0.41	**62.67 ± 0.54**	61.84 ± 1.02
E. time (s)	1710 ± 301	**1213 ± 450**	1590 ± 252	1530 ± 325

Table 3: Risultati ottenuti per il task di *function prediction* sul datset PPI; l'asterisco (*), indica i risultati ottenuti eseguendo l'esperimento nella versione senza le features. Sono stati evidenziati in **grassetto** i migliori risultati per ogni tipologia di esperimento effettuato.

Gli esperimenti eseguiti rispetto alla versione del dataset senza le *features* associate ai nodi, evidenziano una maggiore differenza tra il modello basato su grafi e quello basato su ipergrafi; come sarà più evidente per i risultati di *link prediction*, tale fenomeno può essere spiegato dal fatto che la struttura ad ipergrafo costruita conserva più informazioni relative ai nodi rispetto a quella a grafo.

In questi esperimenti, un'altro risultato rilevante è nel minore tempo necessario per portare i metodi basati su HGCN a convergenza. Infatti dalla figura 3 è possibile osservare come il modello HGCN che lavora sull'ipergrafo costruito mediante JC ha impiegato

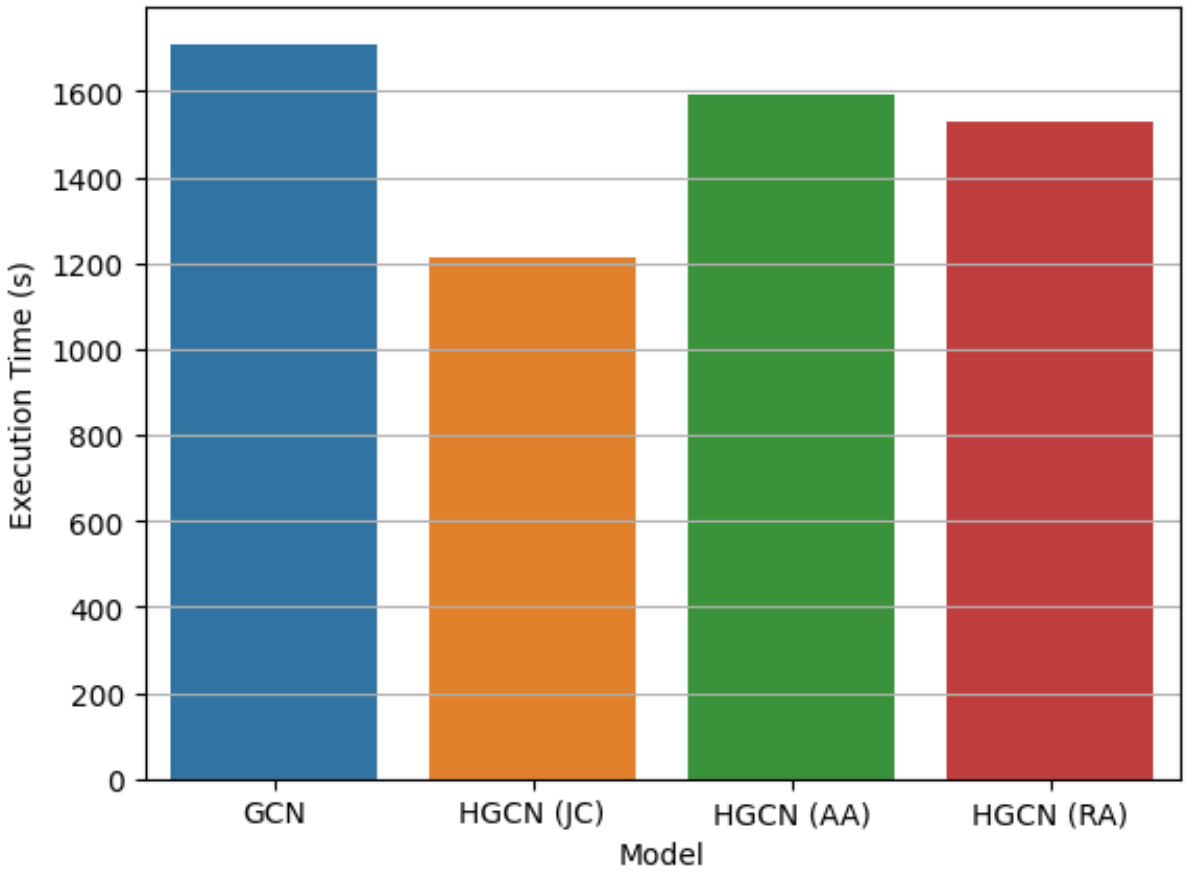


Figure 3: Visualizzazione dei tempi di esecuzione riportati nella tabella 5.

mediamente 1213 secondi per convergere, contro i 1710 secondi medi impiegati dal modello basato su GCN.

Nella maggioranza dei casi, per il task di *function prediction*, i modelli basati su HGCN, hanno permesso di ottenere risultati leggermente più stabili, in termini di varianza, rispetto al modello basato su GCN.

Link Prediction Per il task di link prediction, invece, i risultati mostrano un incremento più deciso tra i modelli che sfruttano le relazioni di alto ordine.

	GCN	HGCN (JC)	HGCN (AA)	HGCN (RA)
ROC	75.70 ± 4.47	**80.60 ± 1.44**	80.36 ± 1.55	80.02 ± 1.53
E. time (s)	**2136 ± 780**	2605 ± 1084	2621 ± 1204	2760 ± 1190
	GCN	HGCN (JC)	HGCN (AA)	HGCN (RA)
ROC (*)	69.94 ± 4.41	82.04 ± 1.76	**82.33 ± 1.60**	82.20 ± 1.39
E. time (s) (*)	**2026 ± 757**	2473 ± 1047	2604 ± 1170	2595 ± 1186

Table 4: Risultati ottenuti per il task di *link prediction* sul datset PPI; l'asterisco (*), indica i risultati ottenuti eseguendo l'esperimento nella versione senza le features. Sono stati evidenziati in **grassetto** i migliori risultati per ogni tipologia di esperimento effettuato.

Negli esperimenti eseguiti per il task di *link-prediction*, il modello basato su GCN ha impiegato tuttavia meno tempo per convergere durante l'addestramento. Tuttavia,

andando ad analizzare il grafico che mostra l'andamento della funzione di perdita durante l'addestramento, è possibile osservare che il modello giunge semplicemente prima in una condizione di *overfitting* rispetto agli altri modelli.

Andando ad analizzare i risultati ottenuti per il task di link prediction nella casistica in cui non vengono prese in considerazione le *features* associate ai nodi, in alcuni casi possiamo osservare un miglioramento di performance anche rispetto alla versione che le usa. Ciò potrebbe indicare, che la conformazione stocastica delle features ha funzionato come tecnica di regolarizzazione, andando a ritardare lo stato di *overfitting* del modello.

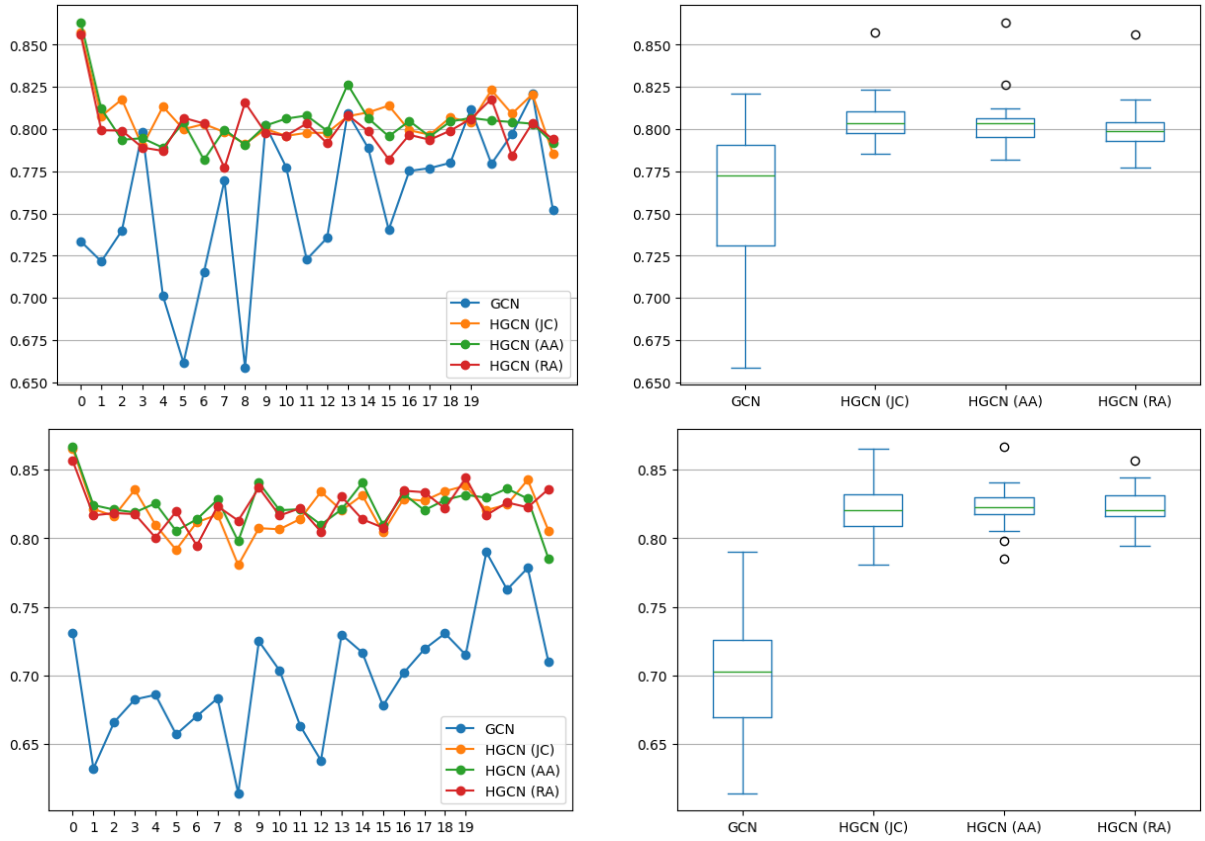


Figure 4: Risultati sul task di link prediction utiizzando i diversi metodi di conversione. In alto sono mostrati i risultati nell'esecuzione del task sfruttando le *features* dei nodi, in basso sono riportati i risultati analoghi nella versione con features inizializzate casualmente.

	GCN	HGCN (JC)	HGCN (AA)	HGCN (RA)
ROC	85.94 ± 0.73	**99.06 ± 0.16**	98.96 ± 0.54	99.03 ± 0.56
E. time (s)	268 ± 17	**105 ± 4**	113 ± 3	108 ± 3
	GCN	HGCN (JC)	HGCN (AA)	HGCN (RA)
ROC (*)	28.56 ± 1.51	98.02 ± 0.84	**98.07 ± 1.37**	97.98 ± 0.84
E. time (s) (*)	262 ± 21	105 ± 2	**101 ± 8**	103 ± 2

Table 5: Risultati ottenuti per il task di *link prediction* sul datset PPI144; l'asterisco (*), indica i risultati ottenuti eseguendo l'esperimento nella versione senza le features. Sono stati evidenziati in **grassetto** i migliori risultati per ogni tipologia di esperimento effettuato.

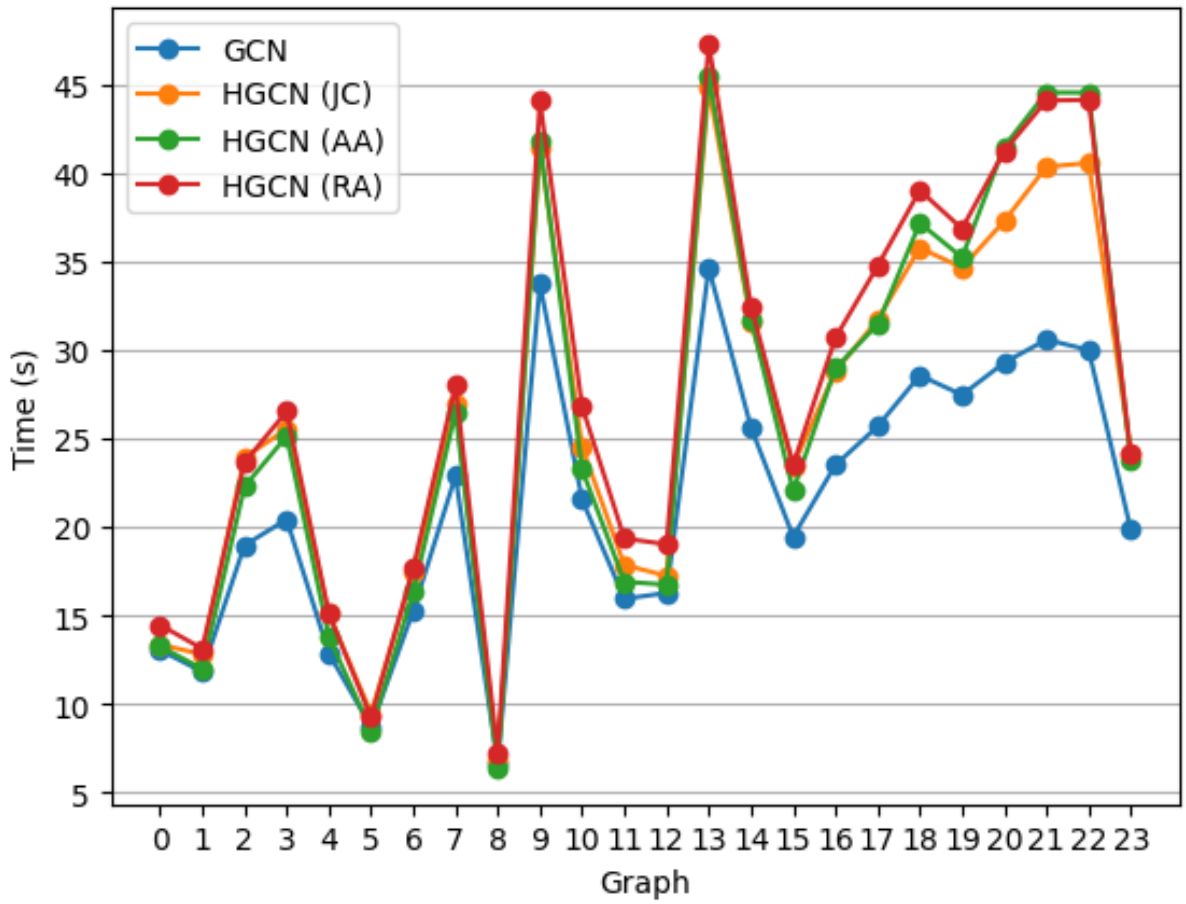


Figure 5: Tempi di esecuzione per ognuno dei 24 grafi sul task di link prediction

6.1 Ulteriori considerazioni

I veri responsabili di questo incremento di prestazioni sono i motif, in particolari le clique, infatti gli ipergrafi sono stati utilizzato solo come strumento per sfruttare queste banche di informazioni in maniera adeguata. Nell'appendice, è riportato lo studio effettuato sul dataset PPI144 riguardo l'agreement presente tra le varie taglie di clique esistenti tra i nodi di questi grafi.

7. Conclusioni

Gli esperimenti effettuati evidenziano quanto i pattern strutturali nascosti nelle reti possano essere banche di informazioni gratuite che la maggior parte dei modelli non può sfruttare. Molti altri lavori, tentano di risolvere i task visti in questo lavoro utilizzando modelli più complessi, con un numero eccessivamente elevato di parametri; tali modelli rappresentano la attuale *baseline* per moltissimi dei problemi che riguardano le reti, tuttavia, l'utilizzo di tali modelli richiede un elevata capacità computazionale, oltre che l'utilizzo di molti iperparametri. In questo lavoro abbiamo quindi dimostrato come sia possibile applicare una particolare tecnica di *data preparation* al fine di costruire strutture più ricche di informazioni, utilizzando modelli più semplici per la risoluzione di diversi

task.

Il task di *link prediction* beneficia maggiormente dell'utilizzo di queste informazioni aggiuntive, permettendo di ottenere risultati nettamente maggiori rispetto alla struttura a grafo originale.

Come possiamo spiegare questo miglioramento? Quello che accade dopo aver lavorato i dataset in questo modo, può essere spiegato attraverso 2 principali possibili motivazioni: (1) i legami che vengono selezionati per costruire gli iperarchi sono "più forti" in relazione all'euristica utilizzata. Tali iperarchi formano una sorta di *mini-cluster* tali che ogni gruppo di nodo ha molto probabilmente caratteristiche simili nella rete. (2) L'operazione *message passing* non è limitata dalla profondità della GCN, piuttosto, con una singola operazione di convoluzione riusciamo a propagare le feature a nodi che normalmente nella struttura a grafo sarebbero distanti.

7.1 Direzioni future

Come accennato in precedenza, i veri protagonisti di questi risultati non sono gli ipergrafi, ma i motif, più in particolare le clique. Grazie alle loro proprietà legate alla significatività, è stato possibile sfruttare queste banche di informazioni per arricchire i nostri dati. Tuttavia, dall'ulteriore analisi effettuata, riportata in appendice, si ipotizza che un possibile miglioramento applicabile ai dati, sta nel ricercare un sottoinsieme di motif più complessi, specifici del dominio dei dati, che permettano di migliorare del performance del modello.

Eseguendo questo studio, sarebbe possibile inoltre ottenere una nuova metrica di *significatività* dei motif dei grafi, basata sul contributo informativo dei motif ad un task piuttosto che sul conteggio di questi speciali sottografi. Nel limite delle ricerche effettuate, al momento della scrittura di questo lavoro, non esistono lavori che hanno condotto tale tipo di studio.

7.2 Lavori correlati

Nel corso degli ultimi anni, con la crescente popolarità degli ipergrafi, diversi lavori hanno proposto uno studio delle reti biologiche attraverso l'uso di questa particolare struttura matematica.

Gli autori di (Murgas et al., 2022) mostrano un ulteriore interessante tecnica per la conversione di una PPI network in un ipergrafo, basandosi sulle interazioni individuate in sottografi con particolari proprietà; Un altro interessate lavoro, è quello di (Lu et al., 2023), nel quale è descritto il processo con cui sono stati costruiti 60 diversi dataset ad ipergrafo attraverso una tecnica simile al precedente, mostrando l'applicazione ed i risultati del task di *hyperlink prediction* su questi. Per uno studio diverso, abbiamo il lavoro di (Lugo-Martinez et al., 2021), il quale presenta un approccio basato su ipergrafi per modellare sistemi biologici, eseguendo test riguardo il task di classificazione dei nodi e *link prediction* su delle particolari viste di queste strutture. Il metodo da loro proposto, si

basa sull'uso di *kernel* applicati a nodi etichettati (o colorati) di un grafo. Hanno testato il loro lavoro su 15 diverse reti biologiche.

Ulteriori due studi sull'applicazione degli ipergrafi alle PPI network sono invece (Xia et al., 2024) e (Z. Zhang et al., 2024): il primo, propone un metodo denominato *HyperGraphComplex* basato su *Hypergraph Variational Autoencoder* che permette di catturare feature espressive dalle sequenze di proteine senza la necessità di una fase di *feature engineering*; il secondo propone una *graph neural netowrk* gerarchica, denominata *HiGPPIM*, per la previsione di modulatori di molecole focalizzandosi nelle PPI netowrk.

References

Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, 25(3), 211–230. https://doi.org/https://doi.org/10.1016/S0378-8733(03)00009-1

Antelmi, A., Cordasco, G., Polato, M., Scarano, V., Spagnuolo, C., & Yang, D. (2023). A survey on hypergraph representation learning. *ACM Comput. Surv.*, 56(1). https://doi.org/10.1145/3605776

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., & Sherlock, G. (2000). Gene ontology: Tool for the unification of biology. https://doi.org/10.1038/75556

Bai, S., Zhang, F., & Torr, P. H. S. (2019). Hypergraph convolution and hypergraph attention. *CoRR*, abs/1901.08150. http://arxiv.org/abs/1901.08150

Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., Young, J.-G., & Petri, G. (2020). Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874, 1–92. https://doi.org/10.1016/j.physrep.2020.05.004

Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. https://arxiv.org/abs/2104.13478

Chang, A., Schomburg, I., Placzek, S., Jeske, L., Ulbrich, M., Xiao, M., Sensen, C. W., & Schomburg, D. (2014). BRENDA in 2015: exciting developments in its 25th year of existence. *Nucleic Acids Research*, 43(D1), D439–D446. https://doi.org/10.1093/nar/gku1068

Graphlets in network science and computational biology. (2019). In *Analyzing network data in biology and medicine: An interdisciplinary textbook for biological, medical and computational scientists* (pp. 193–240). Cambridge University Press.

Greene, C. S., Krishnan, A., Wong, A. K., Ricciotti, E., Zelaya, R. A., Himmelstein, D. S., Zhang, R., Hartmann, B. M., Zaslavsky, E., Sealfon, S. C., Chasman, D. I., FitzGerald, G. A., Dolinski, K., Grosser, T., & Troyanskaya, O. G. (2015). Understanding multicellular function and disease with human tissue-specific networks. https://doi.org/10.1038/ng.3259

Grindrod, P., & Kibble, M. (2004). Review of uses of network and graph theory concepts within proteomics [PMID: 15966817]. *Expert Review of Proteomics*, 1(2), 229–238. https://doi.org/10.1586/14789450.1.2.229

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37, 547–579.

Kipf, T. N., & Welling, M. (2016a). Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907. http://arxiv.org/abs/1609.02907

Kipf, T. N., & Welling, M. (2016b). Variational graph auto-encoders. https://arxiv.org/abs/1611.07308

Lu, Y., Huang, Y., & Li, T. (2023, November). *More is different: Constructing the most comprehensive human protein high-order interaction dataset*. https://doi.org/10.1101/2023.11.06.565906

Lugo-Martinez, J., Zeiberg, D., Gaudelet, T., Malod-Dognin, N., Przulj, N., & Radivojac, P. (2021). Classification in biological networks with hypergraphlet kernels. *Bioinformatics*, 37(7), 1000–1007.

Murgas, K., Saucan, E., & Sandhu, R. (2022). Hypergraph geometry reflects higher-order dynamics in protein interaction networks. *Scientific Reports*, 12. https://doi.org/10.1038/s41598-022-24584-w

Xia, S., Li, D., Deng, X., Liu, Z., Zhu, H., Liu, Y., & Li, D. (2024). Integration of protein sequence and protein–protein interaction data by hypergraph learning to identify novel protein complexes. *Briefings in Bioinformatics*, 25(4).

Xu, J., Wickramarathne, T. L., & Chawla, N. V. (2016). Representing higher-order dependencies in networks. *Science Advances*, 2(5). https://doi.org/10.1126/sciadv.1600028

Zhang, R., Zou, Y., & Ma, J. (2019). Hyper-sagnn: A self-attention based graph neural network for hypergraphs. https://arxiv.org/abs/1911.02613

Zhang, Z., Zhao, L., Wang, J., & Wang, C. (2024). A hierarchical graph neural network framework for predicting protein-protein interaction modulators with functional group information and hypergraph structure. *IEEE Journal of Biomedical and Health Informatics*.

Zitnik, M., & Leskovec, J. (2017). Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14), i190–i198. https://doi.org/10.1093/bioinformatics/btx252

Zitnik, M., Li, M. M., Wells, A., Glass, K., Gysi, D. M., Krishnan, A., Murali, T. M., Radivojac, P., Roy, S., Baudot, A., Bozdag, S., Chen, D. Z., Cowen, L., Devkota, K., Gitter, A., Gosline, S., Gu, P., Guzzi, P. H., Huang, H., ... Milenković, T. (2024). Current and future directions in network biology. https://arxiv.org/abs/2309.08478

Appendice

Nell'appendice sono riportati alcuni esperimenti aggiuntivi con annessa notazione, dello studio dell'*agreement* dei motif nelle reti del dataset PPI144.

Gli esperimenti sono stati eseguiti in ambiente *Ubuntu 22.04*, con CPU *i9 12900K*, *32 GB* RAM e una *RTX 3070 8GB*.

A. Esperimenti aggiuntivi

Per studiare l'esistenza di pattern significativi tra la conta delle clique delle 147 reti del dataset, è stato utilizzato il concetto di distanza tra le *motif signature* di ogni grafo.

Questo approccio si basa sul calcolo della conta normalizzata $S_i(G)$, nel nostro caso, per ogni clique presa in considerazione.

$$S_i(G) = -log\left(\frac{C_i(G)}{\sum_{i=1}^{n} C_i(G)}\right)$$

Successivamente, per calcolare la distanza tra le signature delle conte dei motif di ogni grafo, è stata utilizzata la distanza euclidea, ovvero la misura di distanza che ci ha permesso di ottenere il miglior risultato visivo. Tuttavia, è importante precisare che per produrre un risultato di questo tipo è possibile sfruttare qualsiasi metrica di similarità o distanza, alcuni esempi di queste sono: *cosine similarity*, coefficiente di correlazione di Pearson e simili.

$$D_i(G', G'') = \sum_{i=1}^{n} |S_i(G') - S_i(G'')|$$

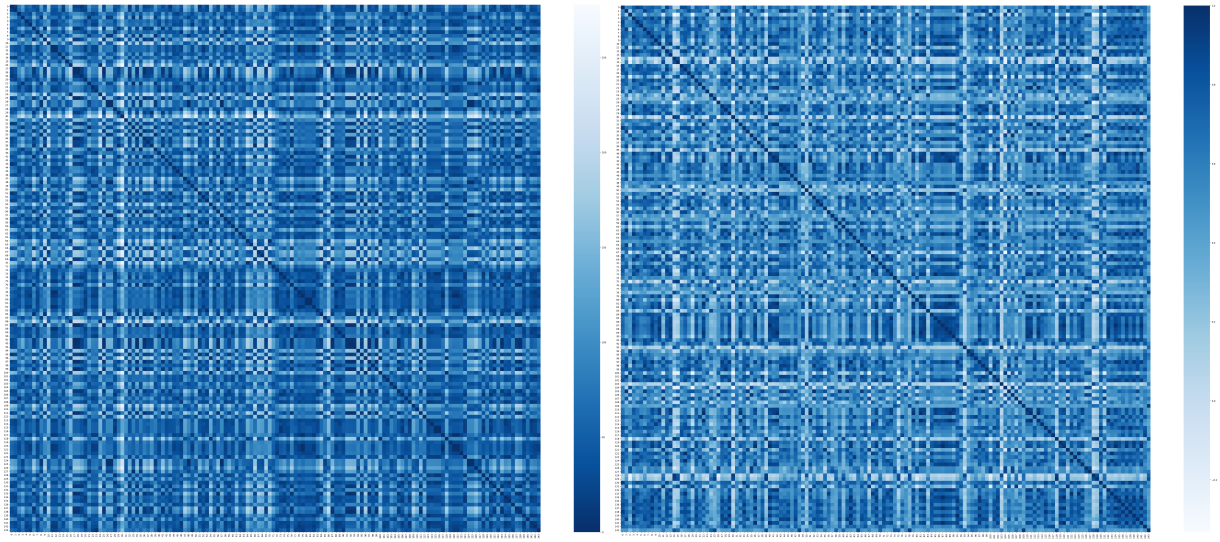
Un altro metodo per calcolare un indice di similarità delle reti è attraverso il profilo caratteristico PC, bisogna innanzitutto calcolare la significance Δ_i dei motif.

$$\Delta_i = \frac{C_i(G) - C_i(G_R)}{C_i(G) + C_i(G_R) + \epsilon}$$

dove ϵ è un valore utilizzato per evitare la divisione per zero. Successivamente possiamo calcolare il profilo caratteristico CP andando a normalizzare i valori di significance.

$$CP_i = \frac{\Delta_i}{\sum_{t \in M} \Delta_t^2}$$

dove M è l'insieme delle tipologie di motif. Grazie a questa analisi, possiamo osservare nella figura A1, che grafi dello stesso *tissue* sono più simili a grafi di *tissue* diverse.



(a) (b)

Figure A1: Nella figura a sinistra, la similarità tra la distribuzione delle clique misurata in tutte le PPI network del dataset PPI144. Nella figura a destra è mostrata la correlazione secondo il coefficiente di Pearson del profilo caratteristico delle clique nello stesso dataset.

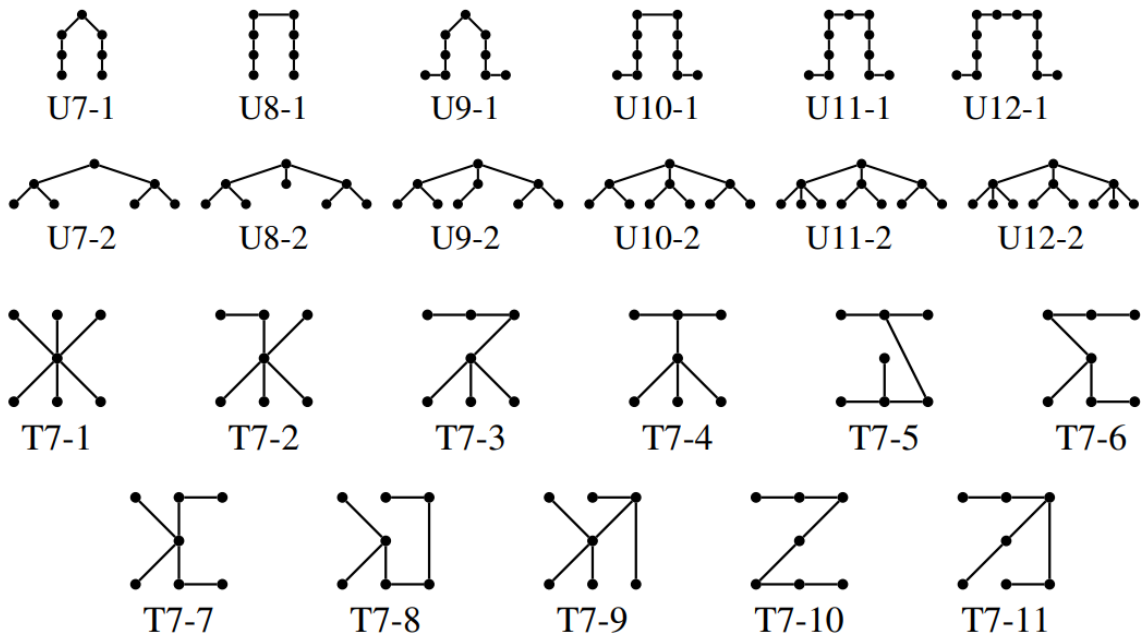


Figure A2: Esempi di tipologie di treelet.

B. Riproducibilità

Per la riproducibilità degli esperimenti, il codice utilizzato è disponibile presso il repository GitHub [1]; è stato utilizzato *Python* nella versione 3.10.14. L'implementazione del modelli basati su deep learning e la pipeline sono state realizzate utilizzando le librerie *PyTorch*, *PyTorch Geometric* e *Scikit Learn*.

[1] https://github.com/strumenti-formali-per-la-bioinformatica/hypergraph-extensions-for-ppi-networks