Relazione Progetto

Francesco Vattiato 1000008830 — Big Data 20/21

6 agosto 2021

1 Introduzione al progetto

Lo scopo del progetto è quello di effettuare delle analisi su pathways attraverso il metodo SEM. Queste analisi permetteranno di comprendere meglio le relazioni tra i vari geni e di sviluppare dei processamenti, basati su tali analisi, per modificare la struttura della pathway.

Le fasi del progetto sono riassimibili nel seguente modo:

- Selezione della pathway e processamento dei dati relativi;
- Calcolo del SEM per ogni triangolo della pathway;
- Filtraggio degli archi.
- Sviluppo di un'interfaccia

Il progetto è stato sviluppato in linguaggio **Python** (versione 3.8), utilizzando le librerie **Pandas**, **Numpy**, **Networkx**, **Semopy**, **pyvis**.

Il materiale provvisto è composto dai seguenti file:

- `controls_counts.tsv`: file contenente i valori di espressione non normalizzati di 5476 geni per 113 controlli.
- `gene_edges.tsv`: file contenente gli archi orientati tra i nodi e il relativo peso che assume il valore di 1 se, dato un arco tra che parte dal nodo A e finisce al nodo B, il gene A stimola l'espressione del gene B. Altrimenti, se il peso è -1, il gene A reprime l'espressione del gene B.
- `pathways.tsv`: file che elenca i nomi di 227 pathway e i relativi archi tra i nodi.

2 Fasi del progetto

In questa sezione descriverò le varie fasi in cui è stato sviluppato il progetto.

2.1 Selezione della pathway e processamento dei dati

La prima fase consiste nella selezione della pathway e del processo dei dati. I dati da processare sono tutte le informazioni riguardo gli archi, i nodi e i triangoli che compongono la rete. Una volta ottenute queste informazioni, si passa alla fase successiva.

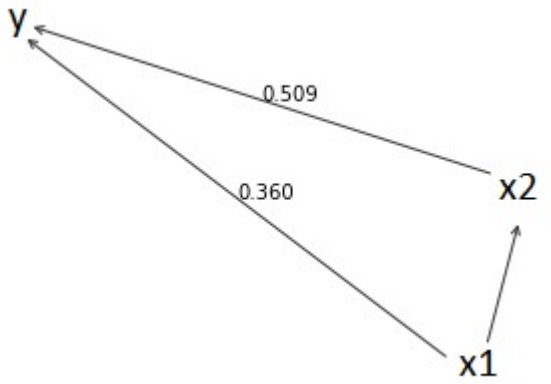

Figura 1: Grafo e fattori calcolati.

2.2 Analisi SEM

In breve, **SEM** è l'acronimo di **S**tructural **E**quation **M**odeling. SEM è la combinazione della Factor Analysis e della multiple regression analysis ed è utilizzata per analizzare le relazioni strutturali tra variabili misurati e variabili latenti. Questo metodo richiede la definizione della struttra su cui addestrare il modello: in questo progetto, per convenzione, ogni triangolo è composto da tre nodi con gli alias x1, x2 ed y, dove x1 è il gene che è collegato direttamente con il nodo target y ed x2 è il nodo che collega indirettamente x1 ad x2.

La libreria utilizzata per tale analisi è semopy e definizione per descrivere tale struttura è la seguente:

$$y \sim x1 + x2 \tag{1}$$

dove il simbolo indica la costruzione di una struttura.

Definita la struttura, il metodo fit del modello verrà applicato alle espressioni dei *tre* geni, ottenendo come risultati i valori dei fattori, il p-value, lo z-value e l'errore. L'esempio di una struttura del genere su cui sono stati calcolati i valori dei fattori è in figura 1

Calcolati i fattori, si esegue un'analisi per verificare se nel triangolo esiste un percorso meno significativo dell'altro. Tale analisi è svolta in questa maniera:

$$if \quad (fact_i < fact_{sum} * 0.1) \tag{2}$$

con $fact_i$ un fattore tra i due calcolati e $fact_{sum}$ è la somma dei due fattori. Il prodotto per 0.1 specifica che il fattore $fact_i$, per essere considerato poco significativo, deve essere minore del 10% di $fact_{sum}$. A seguito della verifica di tale condizione, da verificare su entrambi fattori, sono possibili due scenari:

1. Un cammino viene identificato come significativo e l'altro no. A seguito di questo risultato, si associerà l'etichetta 1 al cammino identificato e l'etichetta 0 all'altro cammino.

2. Entrambi i cammini vengono identificati come "ugualmente" significativi. In questo caso, ad entrambi i cammini verrà associata l'etichetta -1

L'esecuzione del SEM e l'analisi della significatività vengono svolte per tutti i triangoli ed il risultato consisterà in una struttura dati (definita come dizionario in Python) che conserverà per ogni arco l'etichette risultanti dalle analisi svolte sui triangoli di cui fa parte. In poche parole, un arco che collega due nodi A e B e fa parte di cinque triangoli avrà associato cinque etichette, mentre un arco che collega due nodi A e C e che fa parte solo di un triangolo, avrà solo un'etichetta.

2.3 Filtraggio degli archi

Il filtraggio degli archi consiste nel rimuovere gli archi dei triangoli che, a seconda di un criterio, vengono considerati superflui. Tale filtraggio può essere raggiunto con vari approcci che potrebbero differire molto nei risultati finali: tutto ciò rende questa fase quella più delicata.

L'approccio scelto per questo progetto si basa sul bilanciare le etichette -1 e 0 di ogni arco.

E' importante specificare che se un arco è stato etichettato almeno una volta con l'etichetta 1 **non viene fatta nessuna analisi per un eventuale filtraggio.** Questa scelta è motivata dall'idea che se un arco è almeno una volta parte di un cammino predominante, questo non deve essere considerato nel processo di filtraggio perché ha provato di essere **essenziale**.

Dato un arco che collega due nodi A e B, il bilanciamento è così definito:

$$bil_{A,B} = \frac{minus + zeros}{minus \cdot zeros + 1} \cdot \frac{zeros}{minus + 1} \tag{3}$$

con $minus$ il numero di etichette -1 e $zeros$ il numero di etichette 0 dell'arco che collega A e B. Se il valore di $bil_{A,B}$ è elevato, questo verra rimosso.

La prima parte del prodotto serve a bilanciare il numero di $minus$ e $zeros$: il numeratore così fatto evita semplicemente che si abbiano numeri troppo piccoli mentre il denominatore tenderà ad assumere valori bassi quando uno tra $minus$ o $zeros$ è decisamente più grande dell' altro (e.g. $zeros$ uguale a 9 e $minus$ uguale a 1) e assumerà un valore elevato se assumono valori simili (e.g. entrambi uguali a 5). Il primo membro assumerà quindi valori elevati quando è presente un forte squilibrio tra minus e zeros, mentre assumerà valori bassi quando $minus$ e $zeros$ si eguagliano.

Il secondo membro fa in modo che venga pesato maggiormente il valore di zeros rispetto a minus, di conseguenza il filtraggio tenderà a scartare archi dove $zeros$ supera il valore di $minus$. L'equazione così fatta evita il filtraggio di archi che non sono mai stati etichettati con 0. I due $+1$ ai denominatori sono presenti per evitare divisioni per zero.

La variabile $bil_{A,B}$ da sola non basta per decidere se un arco debba essere rimosso o meno: il suo valore va confrontato con uno che faccia da baseline. Tale baseline è così definita:

$$bil_{base} = \frac{2mean}{mean^2 + 1} \cdot \frac{mean}{mean + 1} \tag{4}$$

dove mean è la media data da $\frac{zeros+minus}{2}$. Se $bil_{A,B}$ risulta essere maggiore di bil_{base}, l'arco (A,B) verrà rimosso.

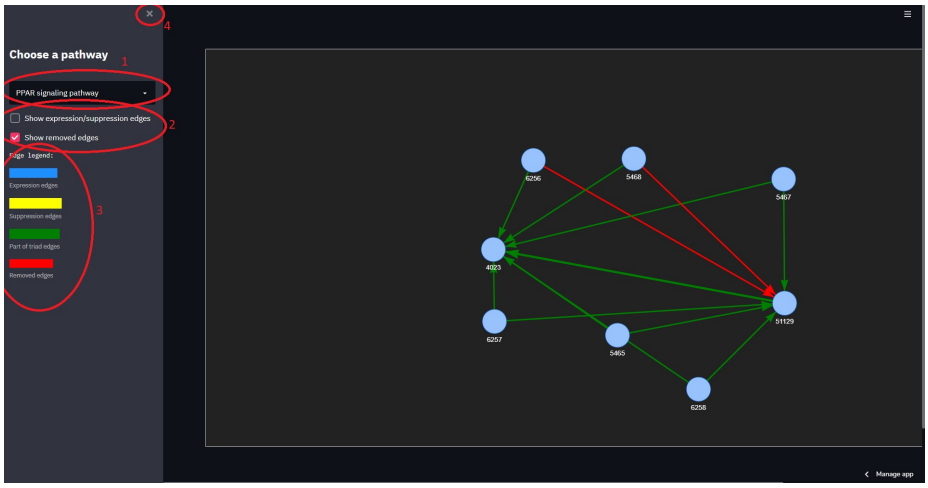

Figura 2: Interfaccia dell' app. In ordine, l'area: 1) Permette di selezionare una pathway tra quelle disponibili; 2) Permette il filtraggio visivo degli archi per categoria; 3)Una semplice legenda per gli archi; 4)Se premuta, rende l'app estesa a tutto lo schermo.

2.4 Implementazione dell'interfaccia

Tutto il lavoro svolto necessita di un'interfaccia che possa esplorare i risultati ottenuti. A tale scopo, ho scelto l'utilizzo della piattaforma **Streamlit**: dato il link ad una repository con codice Python, questa piattaforma prepara le dipendenze e crea la build in modo da ottenere un'app potente ed interattiva. In figura 2 viene presentata l'interfaccia. Ogni volta che viene selezionata una pathway, questa verrà processata seguendo tutti gli step descritti. Nella lista delle pathway sono presenti elementi di cui nodi non sono presenti nel file `controls_counts.tsv`, nel caso in cui venisse selezionata selezionata una pathway di queste, l'applicazione mostrerà un errore che inviterà l'utente a selezionare un'altra pathway.

Per ogni arco, l'interfaccia mostra:

- Un colore che definisce la classificazione dell'arco:
  - Blu se è un semplice arco che stimola l'espressione;
  - Giallo se è un semplice arco che reprime l'espressione;
  - Verde se l'arco fa parte di un triangolo;
  - Rosso se è stato rimosso dal filtraggio.
- Una finestrella passando il mouse su un arco è possibile visualizzare le seguenti informazioni:
  - Se un arco ha almeno un etichetta 1, comparirà scritto "Essential";
  - Expression/Suppression, a seconda del peso, se l'arco è colorato di verde o rosso.
  - equilibrium factor mostra il valore $bil_{A,B}$

Il grafo visualizzato è interattivo ed è possibile spostare i nodi e fare zoom-in/out.

Link all'app di Streamlit: Streamlit App

Link alla repository: Repository

3 Risultati

I risultati finali consisteranno nel filtraggio degli archi meno significativi dei triangoli.

Nella sezione 2.4 si precisa come i risultati necessitano di un'interfaccia per essere interpetrati più facilmente. Grazie all'app sviluppata è possibile visualizzare ciò che l'algoritmo ideato produce.

I risultati variano in base alla struttura della pathway selezionata, quindi non è possibile riportare tutti i possibili riscontri. E' possibile invece mostrare un esempio che dia l'idea di quello che l'algoritmo calcola. In figura 3 viene presentato il processo di filtraggio: l'attenzione è da focalizzare su un sotto-grafo composto dai geni 51129, 6256, 5468, 4023 che formano i due triangoli (6256,51129,4023) ed (5468,51129,4023)

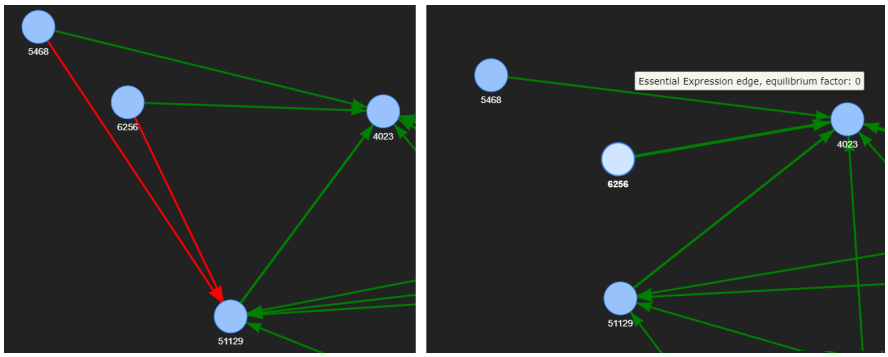

Figura 3: Sottografo della pathway "PPAR signaling pathway" prima e dopo il filtraggio.

Nella prima figura, gli archi (5468,51129) e (6256, 51129), entrambi facenti parte di cammini indiretti al nodo target 4023, vengono classificati come rimovibili. A seguito del filtraggio (seconda figura), entrambi i nodi saranno collegati alla pathway da un solo arco che è proprio quello che li collega al nodo 4023: tali archi, se rimossi, isolerebbero i nodi e questo rappresenta un effetto non voluto. A tal proposito si può notare che, passando il mouse sull'arco che collega i nodi 5468 e 51129, questo risulta essere un arco "Essenziale", quindi nell'analisi SEM ha ottenuto almeno una volta l'etichetta 1. Dato che il filtraggio evita sempre gli archi con queste etichette, i nodi avranno sempre almeno un arco che li collega al resto della pathway. Inoltre, tali collegamenti essenziali danno l'idea di quali siano le vere dipendenze importanti tra i geni. Se un arco non dovesse essere filtrato non implica che questo sia essenziale: semplicemente implica che nelle analisi SEM gli sono stati assegnati solo le etichette -1.

4 Conclusioni

Alla fine di questo progetto, mi ritengo molto soddisfatto del lavoro svolto e del risultato complessivo: nonostante la completezza del lavoro svolto, sono possibili diversi sviluppi di tale lavoro. Uno di questi è sicuramente uno studio statistico sulle etichette assegnate agli archi, in modo tale da evidenziare comportamenti interessanti come dei pattern. Un altro possibile sviluppo è lo studio di un nuovo criterio che funzioni in modo più "intelligente", come ad esempio un criterio basato sulla rimozione greedy degli archi con più etichette a -1 e che alla rimozione degli archi, aggiorni una lista degli archi rimovibili in modo da evitare inconsistenze.