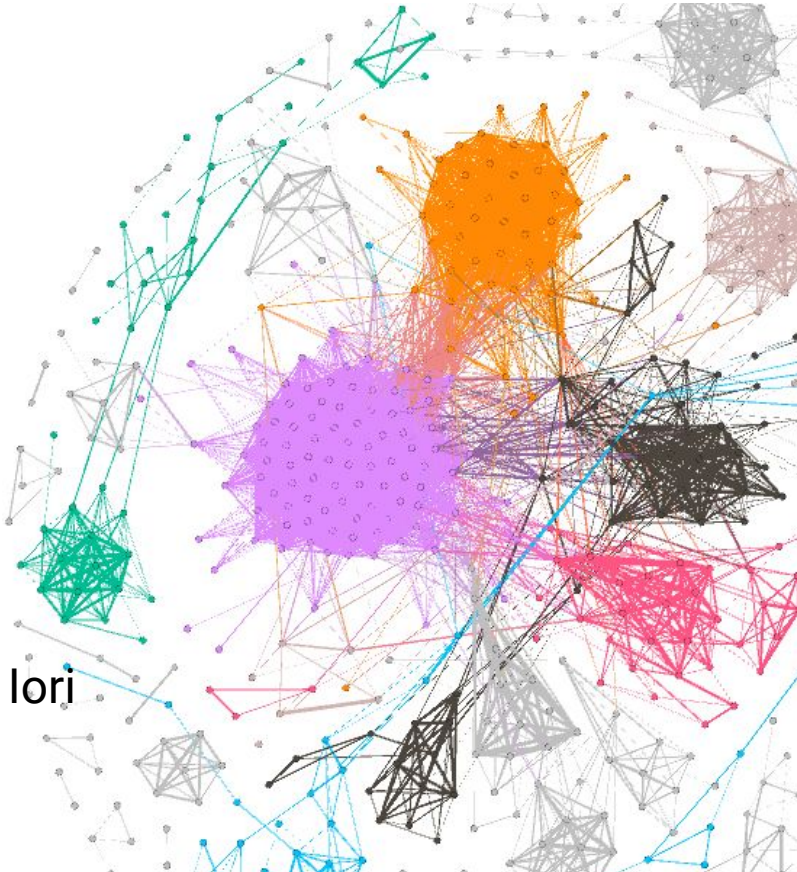
Community Detection

Secondo progetto Big Data

(anno 2016-2017)

Individuare comunità di utenti
sulla base dei gusti musicali



Gruppo: Rido poco
Componenti : Daniel Morales - Alessandro Iori

Che cos'è la Community Detection?

"*Community detection is key to understanding the structure of complex networks, and ultimately extracting useful information from them*" - MIT Edu

La **community detection** è lo **studio** di **reti complesse**, all'interno delle quali è possibile individuare **gruppi** in cui la **membership non** è **esplicitamente data**.

Obiettivo del progetto

1. Individuare comunità di utenti sulla base dei gusti musicali;

2. Confrontare tecniche di Community Detection sul dominio selezionato.

Dataset

LastFM è un **social network** e portale per **l'ascolto** di **musica**. Gli **utenti** possono ascoltare musica, aggiungere tag descrittivi a ciascun **artista**, **seguire** artisti e altri utenti.

GroupLens - LastFM

- 2k utenti
- 18k artisti
- 13k relazioni user-user
- 93k relazioni di ascolto **utente-artista**
- 186k assegnazioni di tag agli artisti

Fasi del progetto

Il progetto si suddivide in quattro fasi principali:

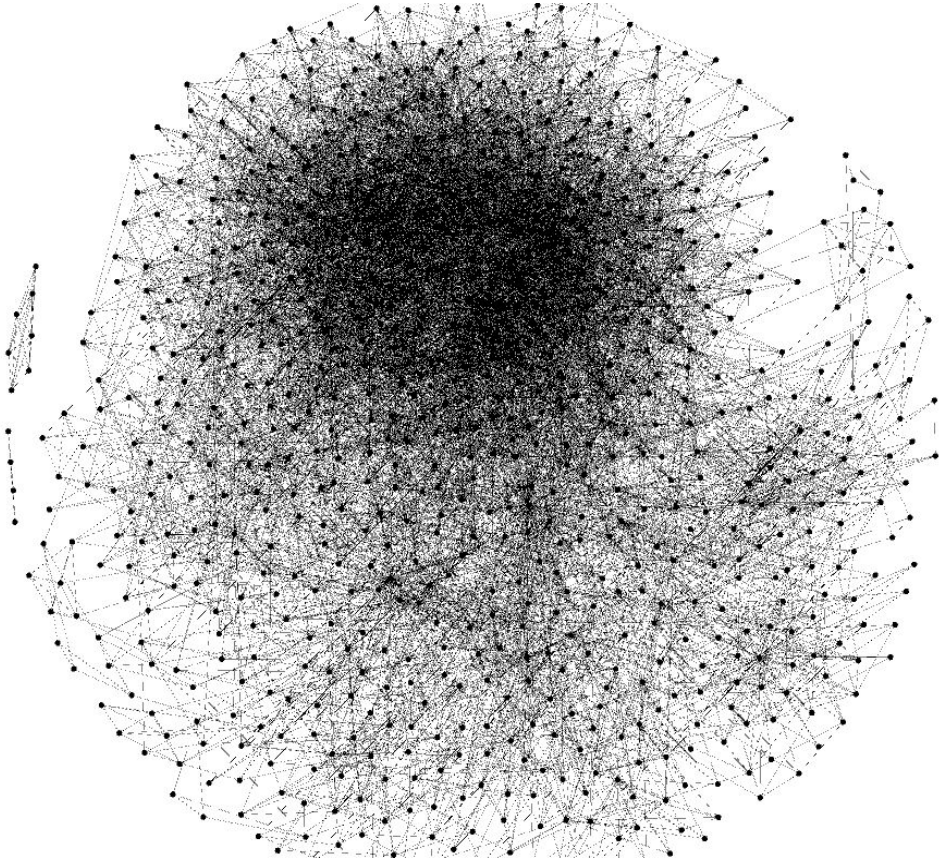
1. Individuazione delle relazioni tra utenti;
2. Individuazione delle community;
3. Modellazione e popolamento graph database;
4. Analisi & Validazione.

Primi passi

- Individuare **comunità** di utenti basandoci su **relazioni** di **amicizia**.
  - Relazione esplicita

- Assunzione (**ERRATA**): Due **utenti amici** (che si seguono) hanno gli **stessi gusti musicali**

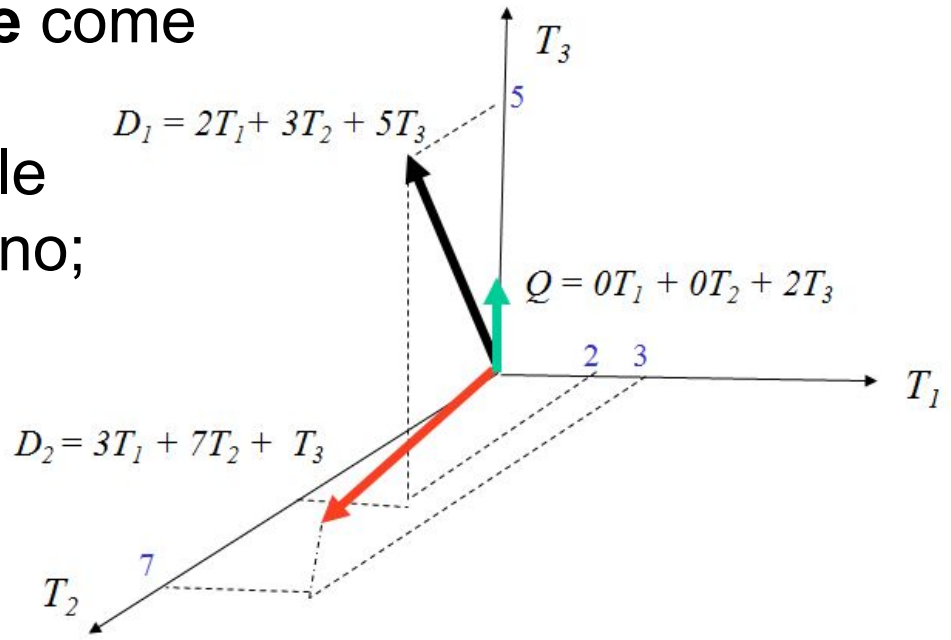
Risultato ottenuto con algoritmo **Clique**



Individuazione relazioni IMPLICITE

L'idea è di creare relazioni fra **utenti** che hanno un **comportamento simile** nel **social network**:

1. Modellato ciascun **utente** come una **feature vector** nello **spazio** multi dimensionale degli **Artisti** che ascoltano;
2. Modellata la **forza** delle **relazioni** tra coppie di **utenti** mediante la **Cosine Similarity**;
3. Ottenuto **grafo di similarità** di utenti. (soglia coseno similarità >= 0.6)



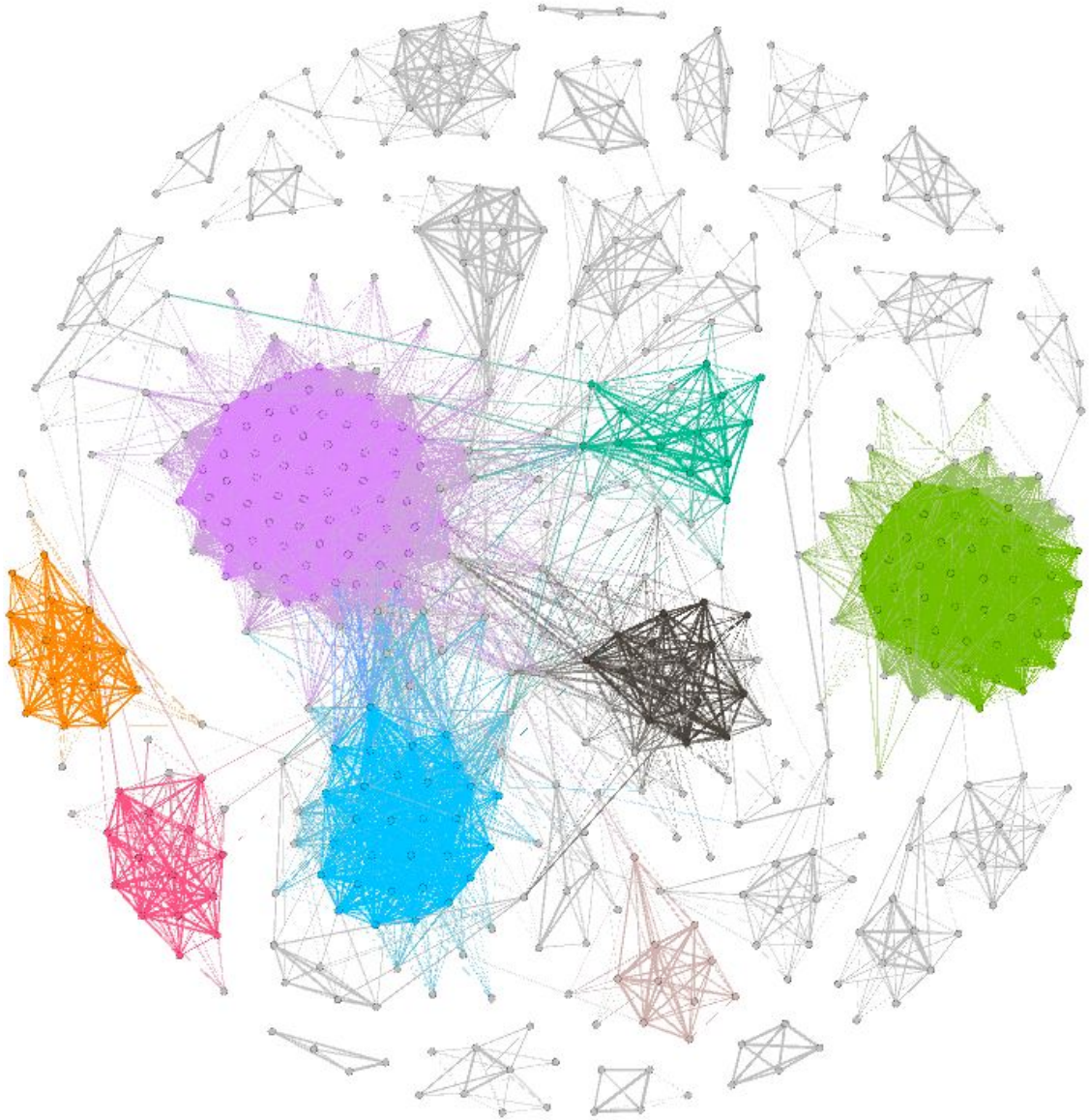
Individuazione delle community: Algoritmi

Le community sono state individuate mediante tre differenti algoritmi di Community Detection selezionati:

- **Clique**
  "*In teoria dei grafi, una cricca (o clique) è un insieme V di vertici in un grafo non orientato G, tale che, per ogni coppia di vertici in V, esiste un arco che li collega.*" - Wikipedia
- **K-plex**
  "*A k-plex is a maximal subgraph with the following property: each vertex of the induced subgraph is connected to at least n-k other vertices, where n is the number of vertices in the induced subgraph*" - Analytictech
- **Louvain Method**
  "*The inspiration for this method of community detection is the optimization of Modularity as the algorithm progresses. Modularity is a scale value between -1 and 1 that measures the density of edges inside communities to edges outside communities.*" - Wikipedia

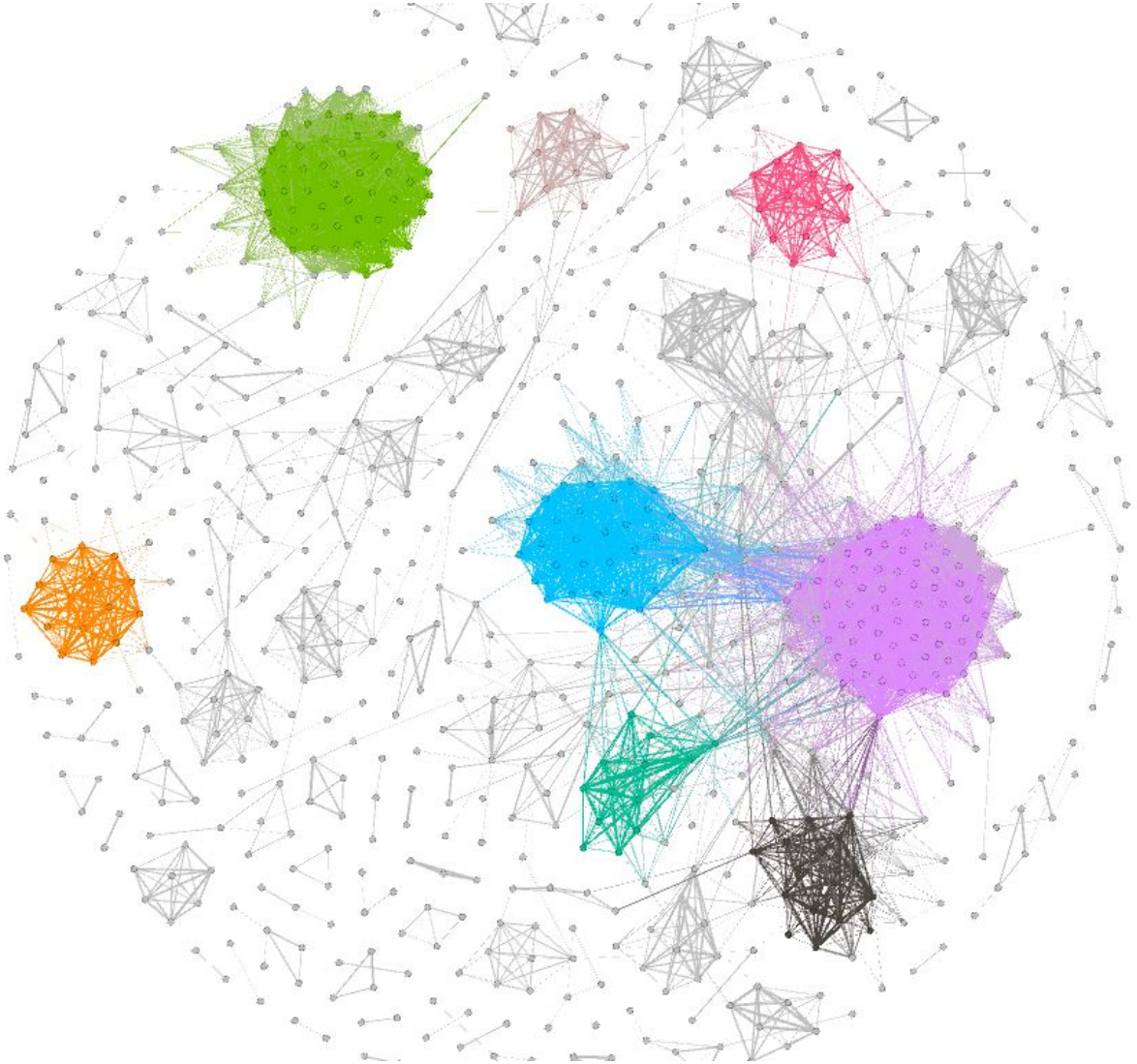
Individuazione delle community: Primi risultati (1)

Clique



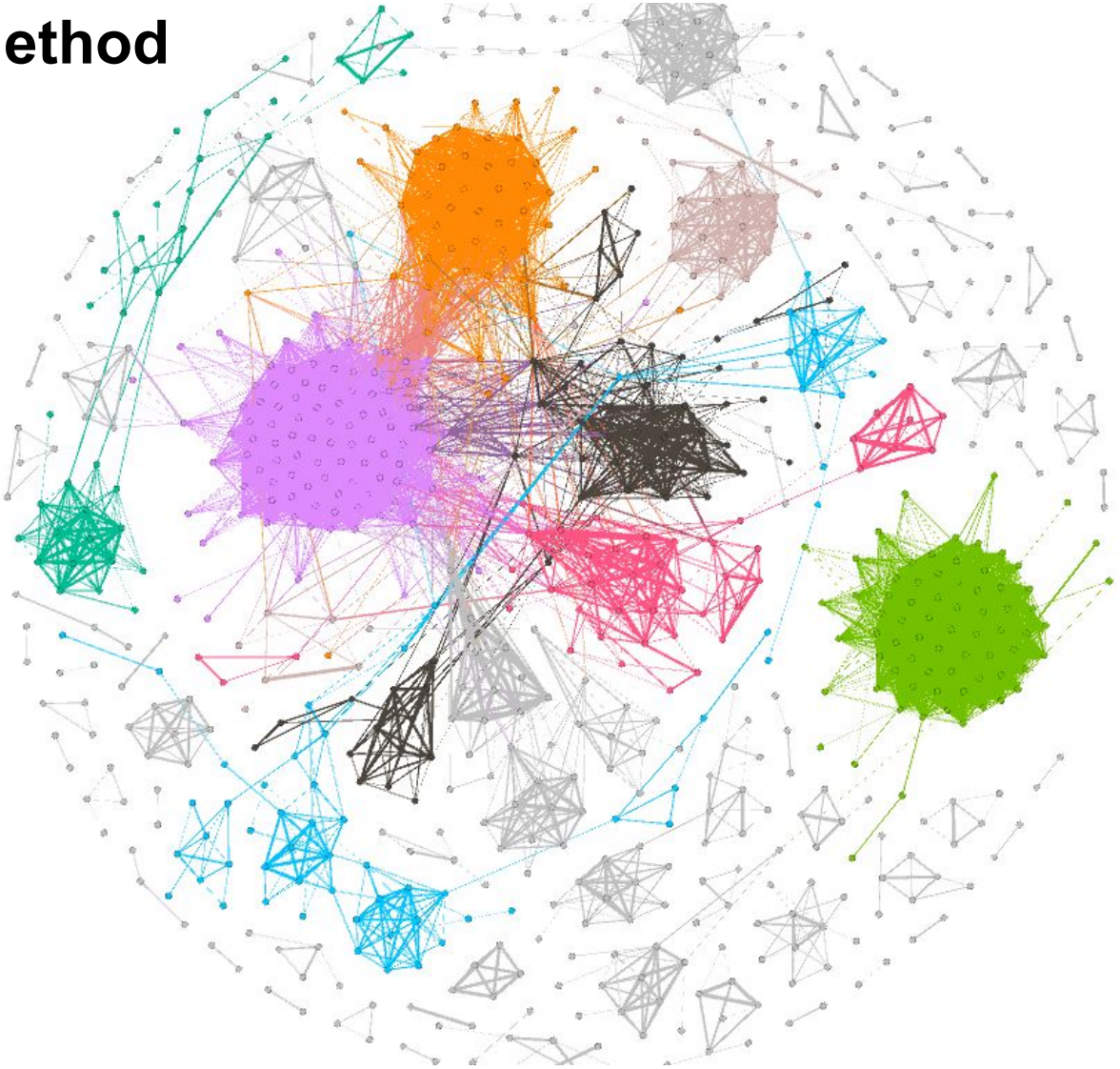
Individuazione delle community: Primi risultati (2)

K-plex

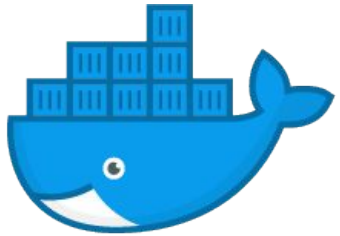


Individuazione delle community: Primi risultati (3)

Louvain Method



Tecnologie: Fase 1 e 2



docker

- **Fase 1: Relazioni** (cosine similarity)
  - Cassandra
  - Spark
  - MLib
  - Docker

- **Fase 2: Community**
  - NetworkX (Python3)
    - Clique
    - Louvain Method
  - K-plex lib (Standford University)
  - Amazon EC2
    - EC2
    - S3 bucket

Analisi dei dati

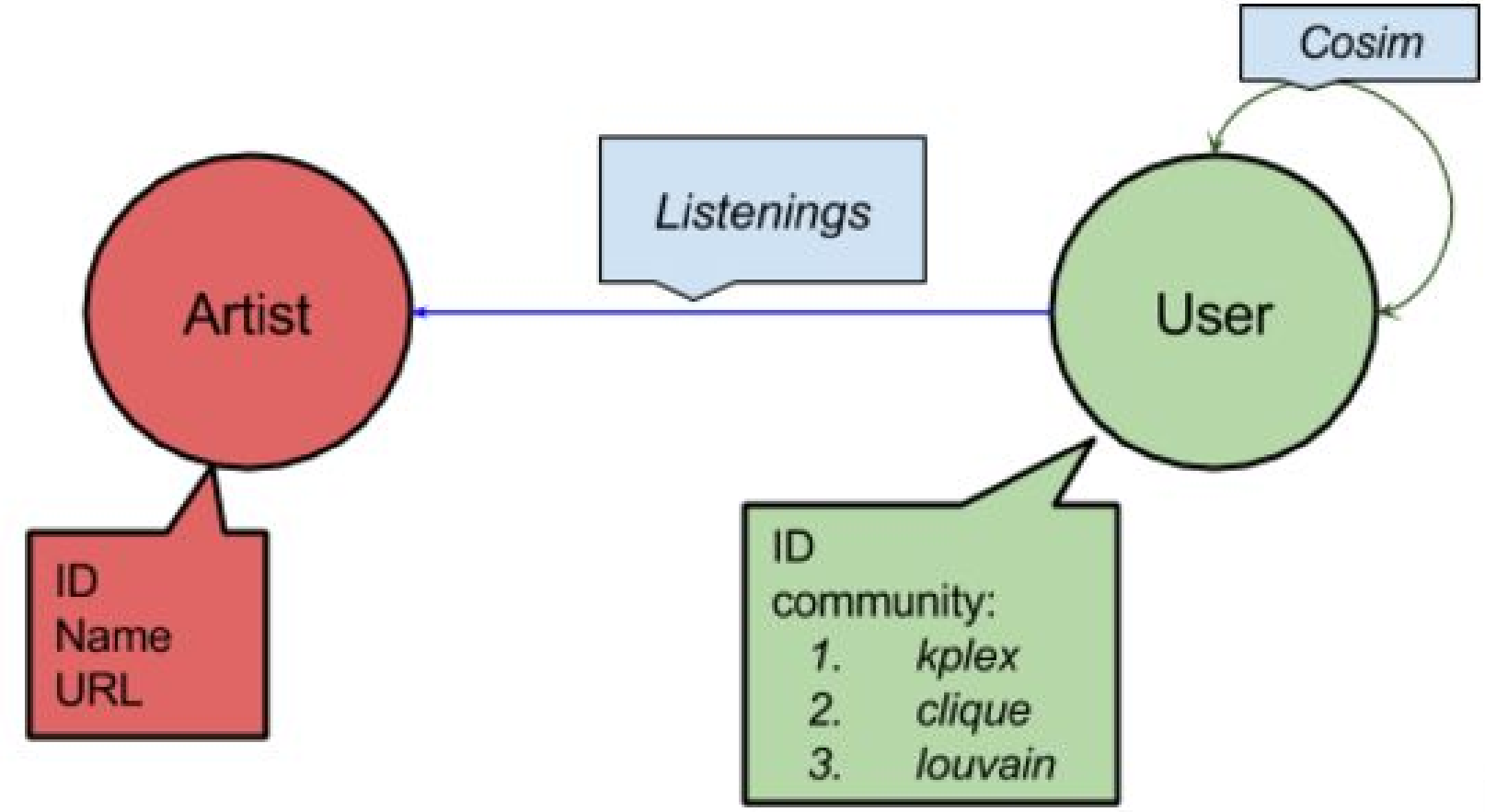
Dati a disposizione dopo la fase 2:

- **Relazioni** tra entità **User** e **Artist**:
  - peso relazione: **numero di ascolti**.

- **Relazioni** tra entità **User** e **User**:
  - peso relazione: valore **coseno similarità**.

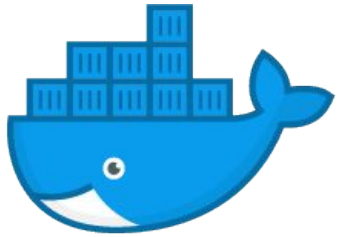
- **Comunità** composte da ID di User, per tipologia:
  - Clique
  - K-plex
  - Louvain

Modellazione dei dati



Graph Database

Motivati dalla complessità delle **relazioni** tra **entità** e dalla **struttura** del **dataset**, è stato selezionato un Graph Database per la memorizzazione: **Neo4j**



- Esecuzione in contenitore **Docker** in ambiente locale;
- Popolamento e interrogazioni mediante query **Cypher** immerse all'interno di script ad-hoc in linguaggio **Python**;

- Flusso popolamento:
  a. User, Artist e relazioni di Listening;
  b. Relazioni cosine similarity tra utenti;
  c. Comunità delle 3 tipologie (clique, k-plex, louvain) per ciascun User.

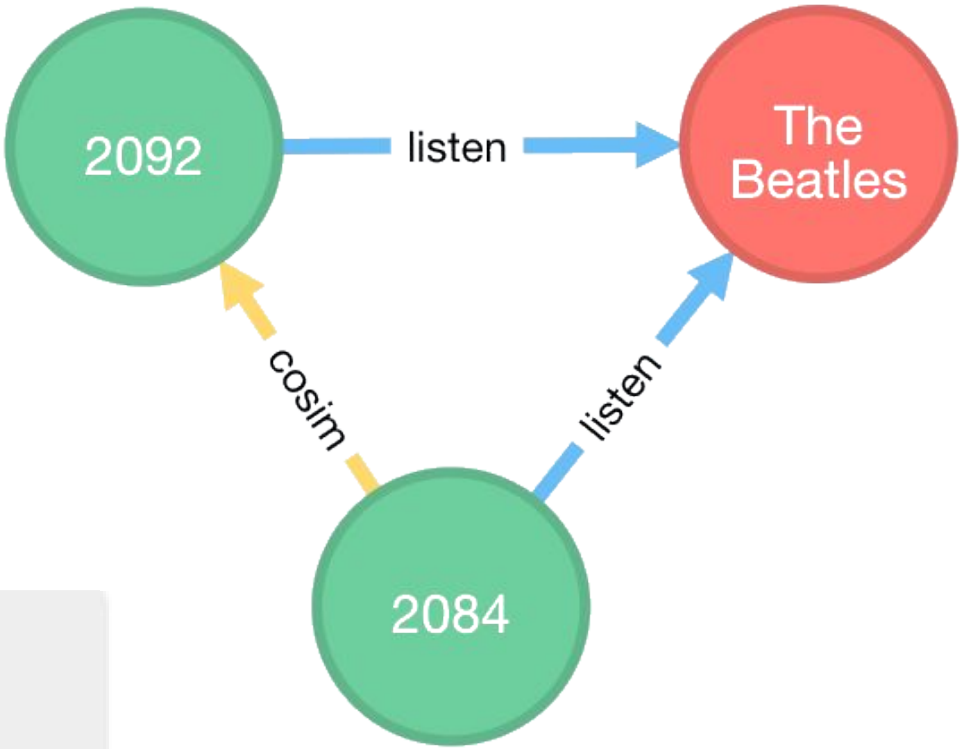
Risultato popolamento

User

kplexes	[346, 625, 626]
cliques	[142, 332]
louvains	[8]
userId	2084

Artist

name	The Beatles
artistId	227
url	http://www.last.fm/music/The+Beatles



Cosim

weight	0.8173175240835568

Listen

weight	891

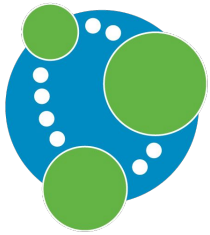
Ottimizzazioni interrogazioni Cypher

Per ottimizzare le query su **User** e **Artist**, sono stati creati due **indici** sulla proprietà **userId** e **artistId** delle due entità:

```
$ CREATE INDEX ON :User(userId)
```

```
$ CREATE INDEX ON :Artist(artistId)
```

Analisi dei risultati



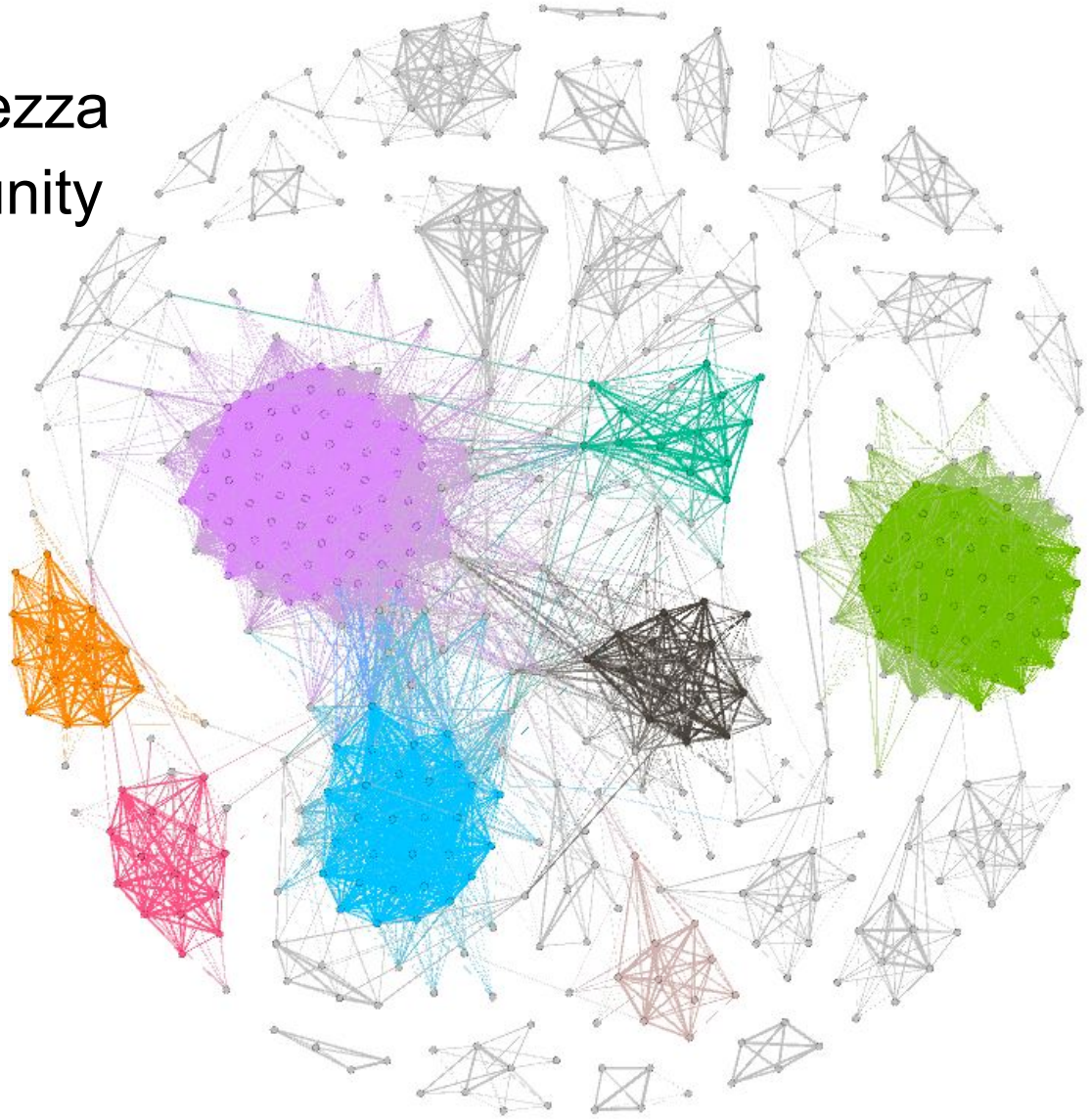
- **Validazione** delle **comunità** prodotte dai diversi algoritmi **verificando** che riflettano effettivamente i **gusti musicali** degli **User**.
- Le **comunità** di **User** individuate dai 3 algoritmi (clique, k-plex, louvain) sono state **selezionate** visibilmente su **Gephi** e analizzate singolarmente attraverso delle query **Cypher** su **Neo4J** aggiungendo gli **Artisti** ascoltati.

```
1 MATCH p=(u1:User)-[r1:cosim]-(u2:User)
2 MATCH q=(u1:User)-[r2:listen]->(a:Artist)
3 MATCH f=(u2:User)-[r3:listen]->(a:Artist)
4 WHERE 3 IN u1.kplexes AND 3 IN u2.kplexes AND toFloat(r1.weight) >=0.8
5 AND toInt(r2.weight) >= 400 AND toInt(r3.weight) >= 400
6 return p, q, f
```

Analisi dei risultati: Clique

Per verificare la correttezza dell'algoritmo di community detection **clique**, abbiamo selezionato un campione di **comunità**:

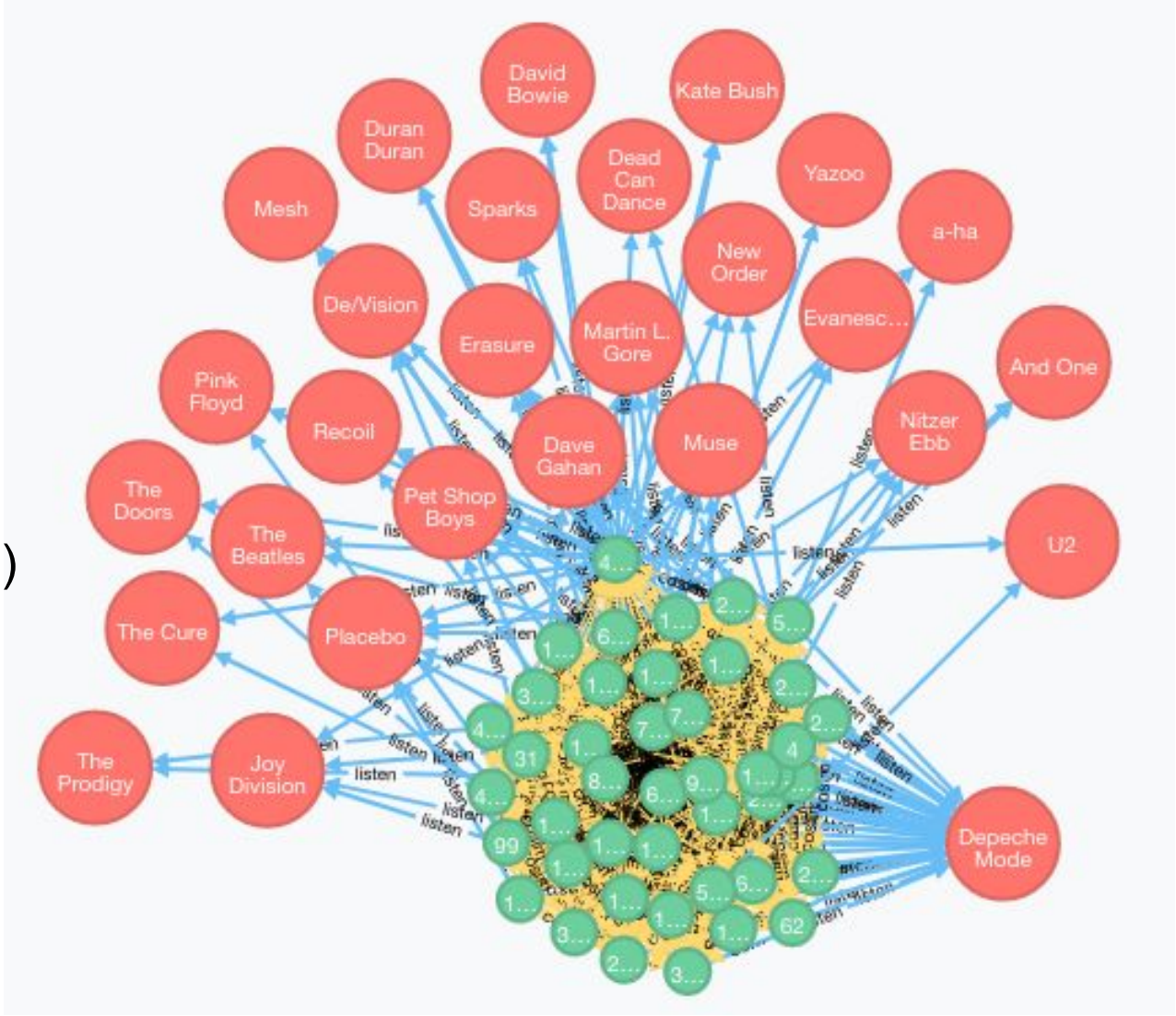
- verde : id 30
- rosa : id 1



Analisi dei risultati: Clique (1)

Community:

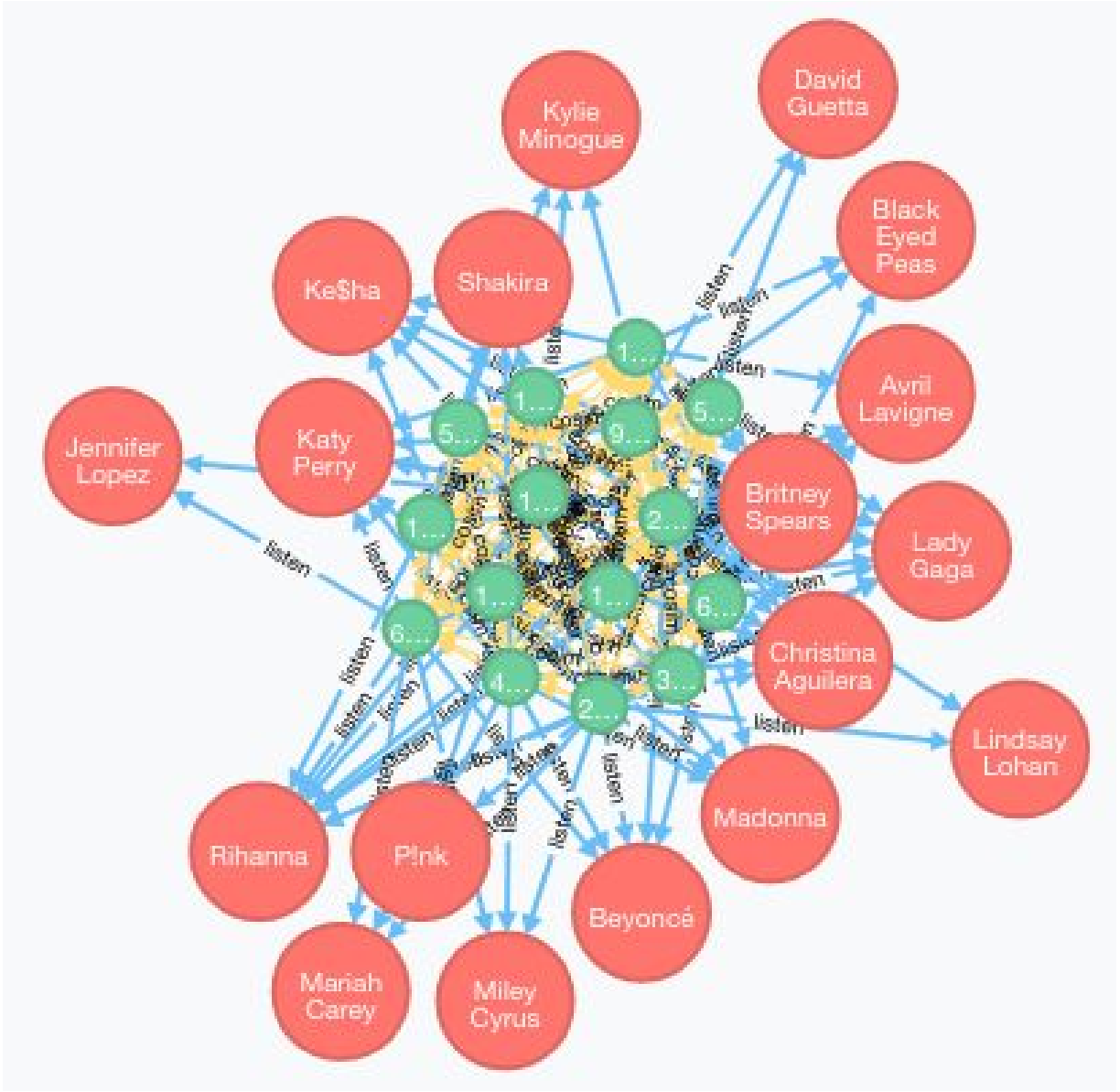
- clique
- id 30
- genere ascoltano **Rock** (prevalentemente)



Analisi dei risultati: Clique (2)

Community:

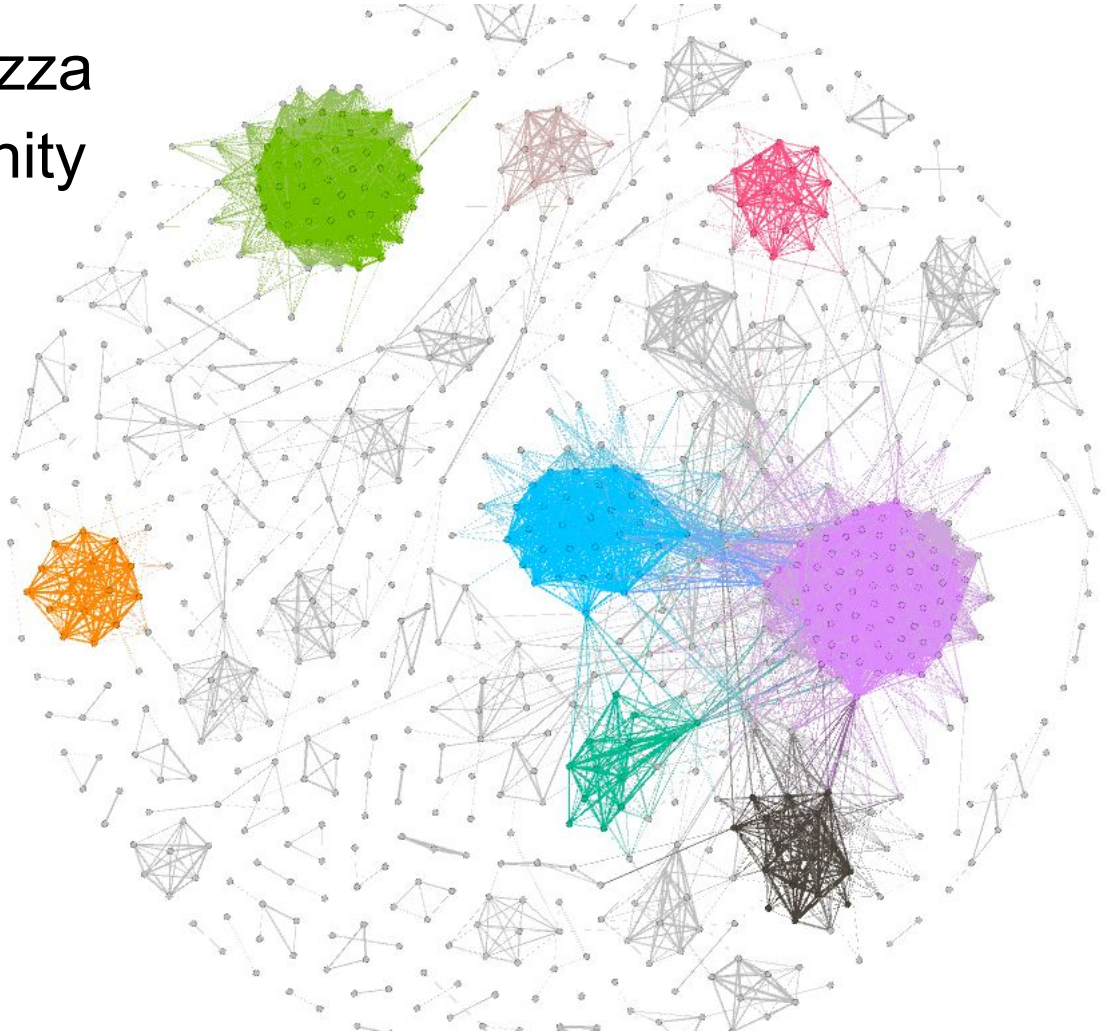
- clique
- id 1
- genere ascoltano **Pop** (prevalentemente)



Analisi dei risultati: K-plex

Per verificare la correttezza dell'algoritmo di community detection **k-plex**, abbiamo selezionato un campione di **comunità** :

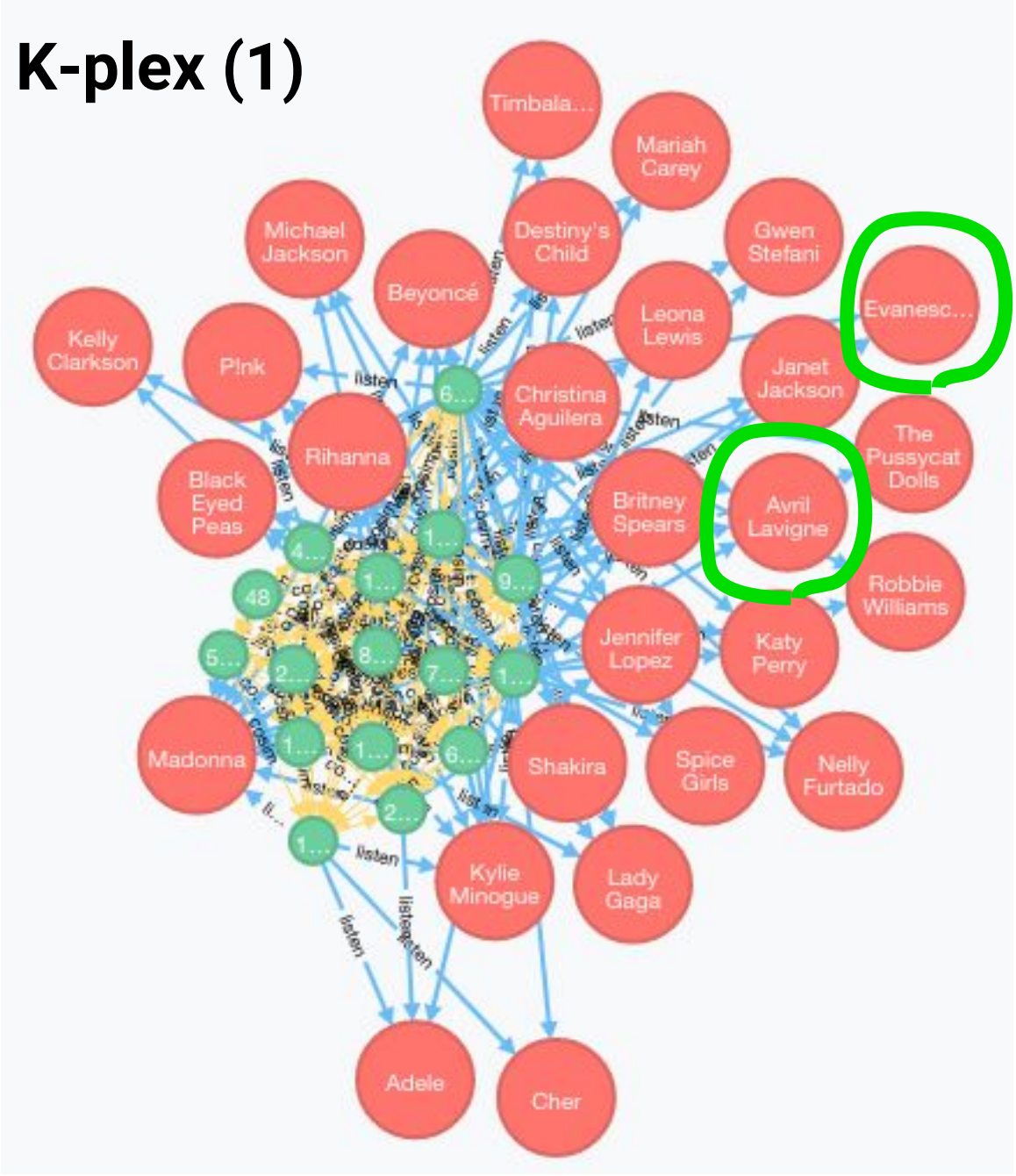
- **nero** : id 4345
- arancione : id 5142



Analisi dei risultati: K-plex (1)

Community:

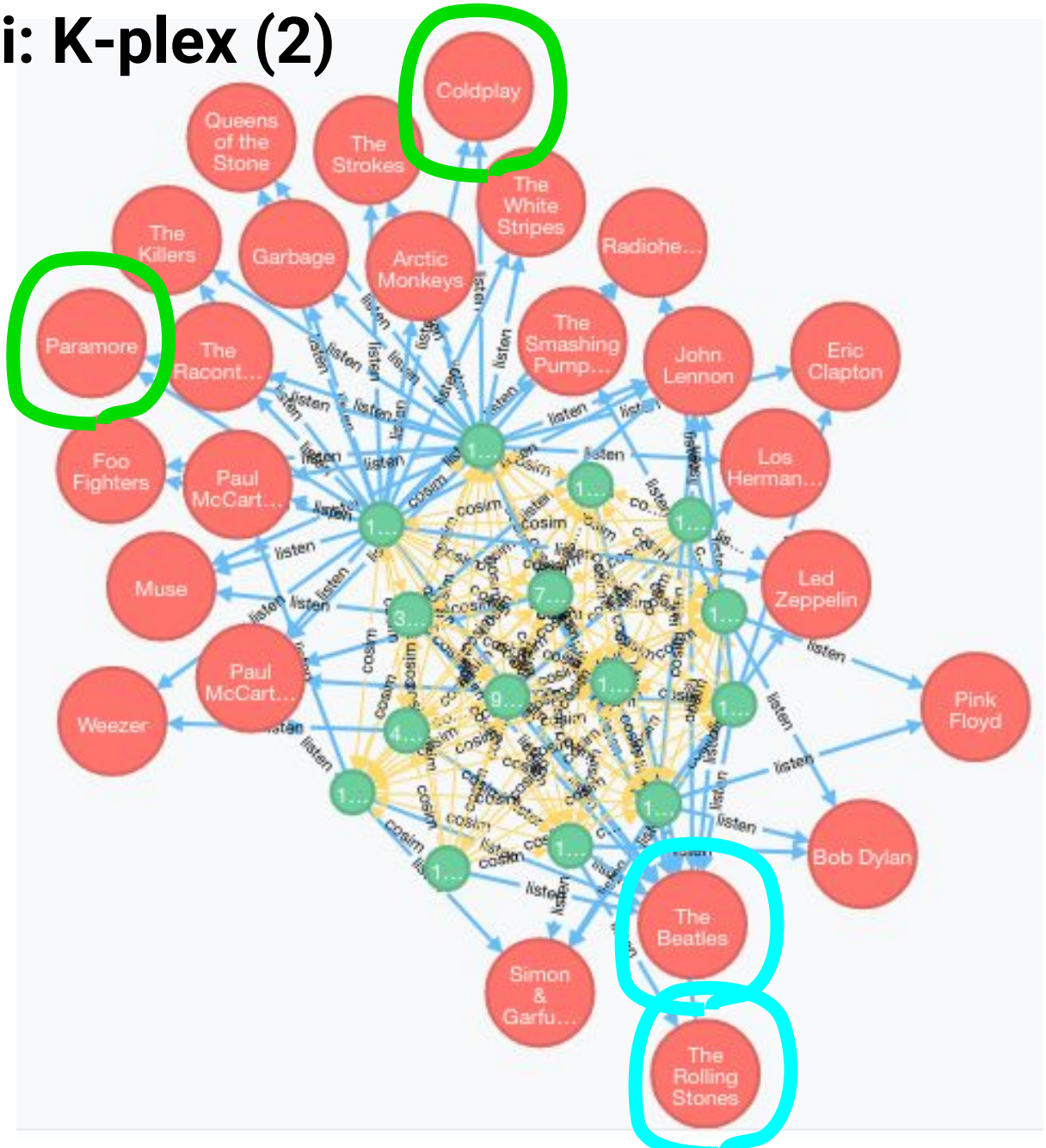
- k-plex
- **nero** id 4345
- genere ascoltano
  Pop
  (prevalentemente)
  Rock e Pop Rock



Analisi dei risultati: K-plex (2)

Community:

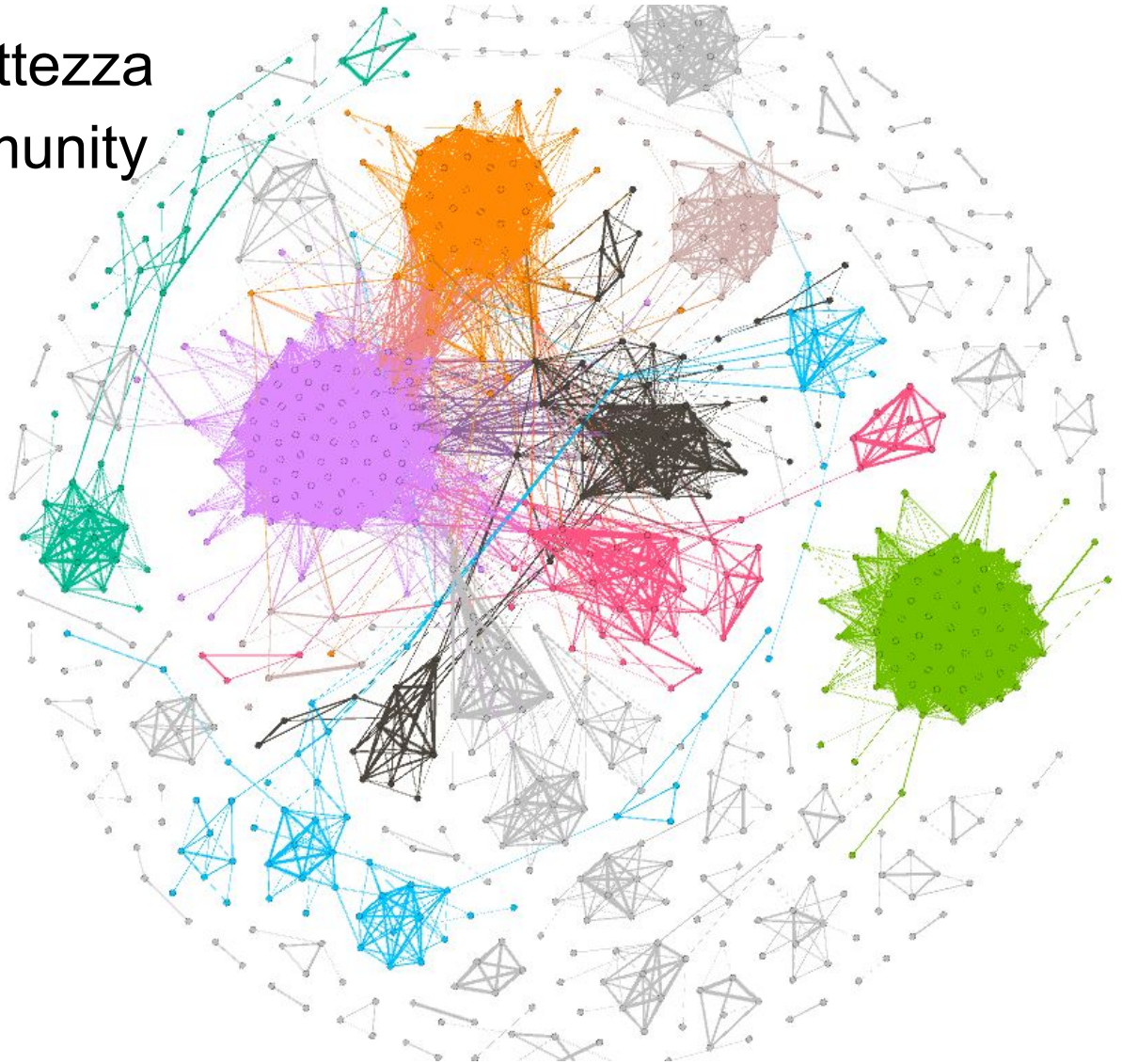
- k-plex
- arancione id 5142
- genere ascoltano **Rock** (prevalentemente) e **Rock Pop**



Analisi dei risultati: Louvain Method

Per verificare la correttezza dell'algoritmo di community detection **louvain**, abbiamo selezionato un campione di **comunità**:

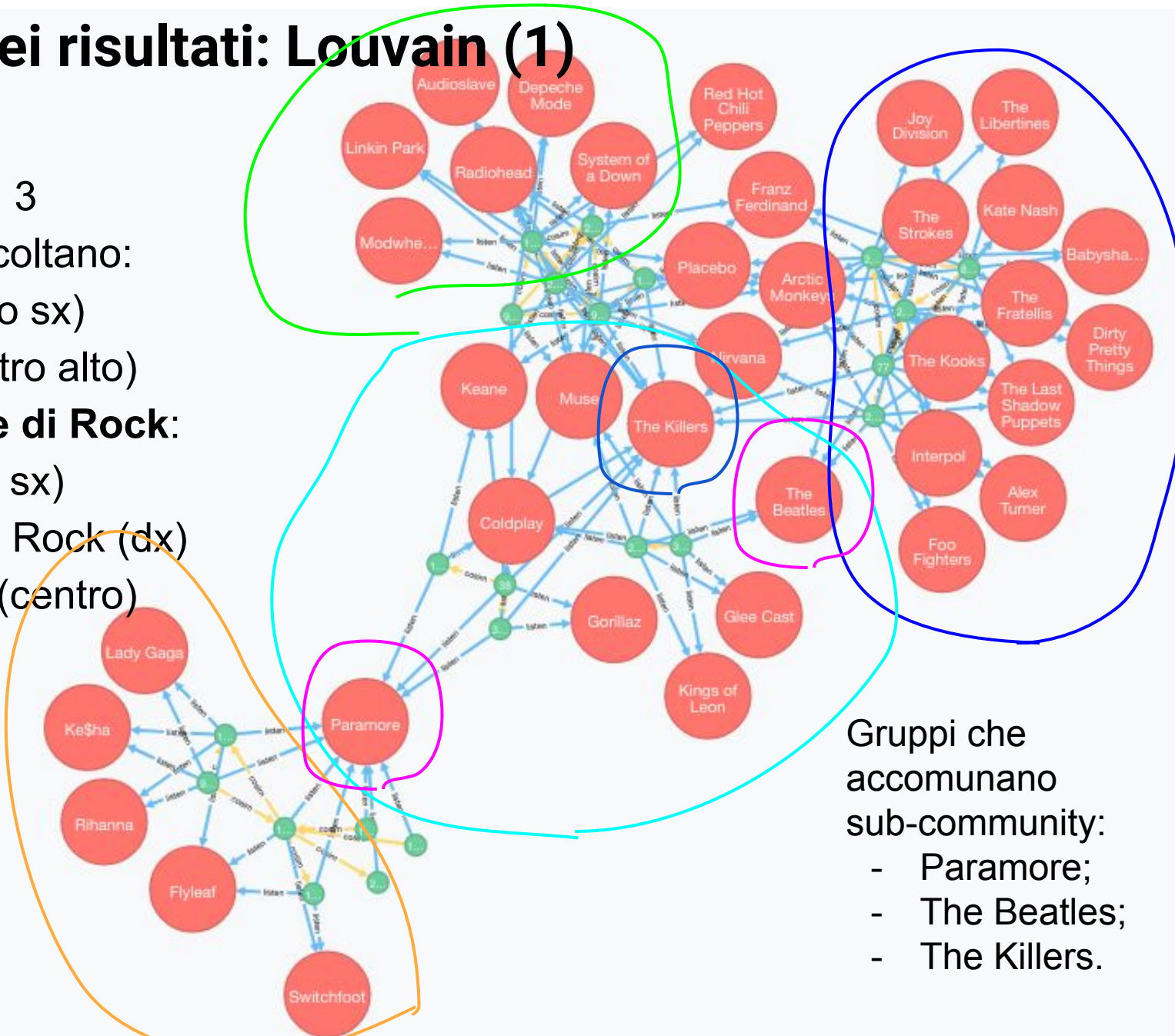
- verde-acqua : id 3



Analisi dei risultati: Louvain (1)

Community:

- louvain, id 3
- genere ascoltano:
  Pop (basso sx)
  Rock (centro alto)
- **Sfumature di Rock**:
  Metal (alto sx)
  Alternative Rock (dx)
  Pop Rock (centro)



Gruppi che accomunano sub-community:

- Paramore;
- The Beatles;
- The Killers.

Combinazione degli algoritmi di community detection

Combinati i risultati dei tre algoritmi su ciascuna **coppia** di **nodi** User **i** e **j** del **grafo**, ottenendo una **matrice NxN** contenente per ciascun elemento una tripla **[h,p,q]** che identifica:

- **h**: numero di community **clique** a cui appartiene la coppia i e j;
- **p**: numero di community **k-plex** a cui appartiene la coppia i e j;
- **q**: appartenenza o meno della coppia i e j a una comunità **louvain**.

Individuate 8 casistiche per ciascun elemento della matrice, che per semplicità descriviamo in binario, indicando con 1 il fatto che la coppia di nodi i e j partecipa almeno ad una comunità dell'algoritmo k-esimo, 0 altrimenti.

Casistiche elemento matrice User

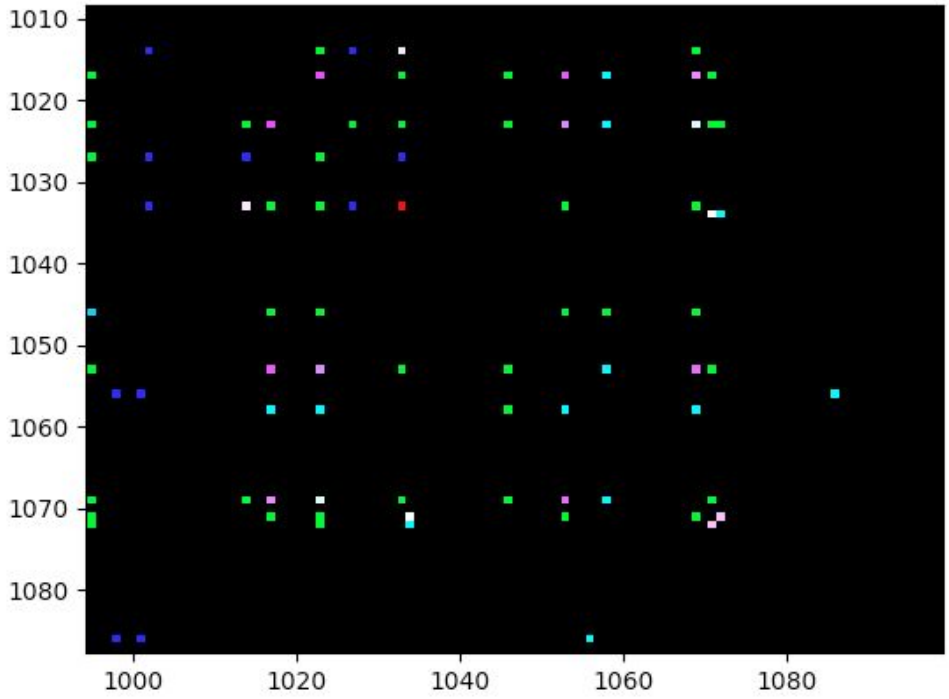
	h	p	q	
1.	0	0	0	Utenti **i,j** non partecipano insieme a comunità.
2.	0	0	1	Utenti **i,j** partecipano insieme a comunità louvain.
3.	0	1	0	Utenti **i,j** partecipano insieme a comunità k-plex.
4.	0	1	1	Utenti **i,j** partecipano insieme a comunità k-plex e louvain.
5.	1	0	0	Utenti **i,j** partecipano insieme a comunità clique. Identifica errore nell'individuazione della comunità da parte di k-plex o clique.
6.	1	0	1	Utenti **i,j** partecipano insieme a comunità clique e louvain. Identifica errore nell'individuazione della comunità da parte di k-plex o clique.
7.	1	1	0	Utenti **i,j** partecipano a comunità clique e k-plex. Occorre prestare attenzione alle cardinalità.
8.	1	1	1	Utenti **i,j** partecipano a comunità clique, k-plex e louvain. Occorre prestare attenzione alle cardinalità di clique e k-plex.

1. e 8. casistiche "forti", 5. e 6. possibile errore algoritmi

Visualizzazione RGB della Matrice

Codice colori:

- **Rosso**:
  solo/prevalentemente clique;
- **Verde**:
  solo/prevalentemente k-plex;
- **Blu**:
  solo/prevalentemente louvain;
- **Celeste**:
  prevalentemente k-plex e louvain;
- **Rosa/Fuxia**: tutti e tre;



Conclusione

Individuate comunità basate sul genere musicale.

Possiamo affermare che:

1. **Clique**: prevalentemente comunità di un singolo genere;
2. **K-plex**: comunità di generi con maggiore flessibilità;
3. **Louvain**: consente di **individuare** in una singola comunità **più generi** (e sottogeneri).
    a. Presenza di **artisti** nelle comunità che hanno il ruolo di **ponte tra generi** e **sottogeneri** diversi ma correlati.

Thanks

/alessandroiori/community-detection-lastfm