UNIVERSITÀ
DI TRENTO

Data Mining 2025/2026

Corrao Jacopo
Matricola: 258412

Professore: Scettri Giacomo

Indice

1 Introduction

The music industry typically measures success through sales charts and streaming numbers (popularity). However, cultural impact is structurally different from popularity. A "one-hit wonder" might sell millions, but an obscure funk track from the 1970s might be sampled by hundreds of hip-hop artists, forming the backbone of an entire genre. The objective of this project is to shift the analysis from volume to structure. By modeling the music ecosystem as a **Directed Graph** where nodes are songs and edges represent sampling relationships we aim to:

- Identify the structural "ancestors" of modern music (Authority).
- Detect communities of influence that transcend traditional genre labels.
- Analyze the flow of influence using distributed graph algorithms.

2 Data Preparation

2.1 Dataset Source

MusicBrainz Data was sourced from the **MusicBrainz** Database, an open music encyclopedia that collects user-contributed metadata. Unlike simplified datasets found on platforms like Kaggle, **MusicBrainz** provides raw PostgreSQL database dumps. The complexity of this dataset lies in its highly normalized structure (Third Normal Form), which requires significant data engineering to reconstruct meaningful relationships. We utilized the core tables: `recording` (songs), `artist_credit` (artist names), `l_recording_recording` (relationships), and the bridging table `link`, which connects specific relationships to their semantic definitions found in the `link_type` table.

2.2 Data Aquisition and engineering

The raw data consisted of tab-separated value (TSV) files without headers, extracted from the `mbdump.tar.bz2` archive. The transformation process from raw relational tables to a structured property graph involved three key phases: Schema Definition, Projection, and Graph Reconstruction.

2.3 Computational Framework Selection

Before addressing the data schema, it is necessary to justify the architectural choice. Given the relational complexity of the MusicBrainz database (highly normalized) and the potential size of the graph, traditional single-node tools like Pandas would be inefficient for the required multi-way joins and iterative computations.
Apache Spark (PySpark) was chosen as the framework to leverage:

- **Distributed Processing**: To handle data ingestion and transformation in parallel.
- **Lazy Evaluation**: To optimize the execution plan of complex transformation pipelines.
- **Iterative Computation**: Essential for implementing graph algorithms like PageRank from scratch.

2.4 Schema-on-Read Definition

Having established the framework, the first challenge was ingestion. Since the raw files lacked headers, relying on automatic schema inference was computationally

expensive and error-prone (risking incorrect data type assignment). I manually defined the StructType for each table based on the MusicBrainz documentation. This ensures strict typing and robust data ingestion.

```
        recording_schema = StructType([
            StructField("id", IntegerType(), True),
            StructField("gid", StringType(), True),
            StructField("name", StringType(), True),
            StructField("artist_credit", IntegerType(), True),
            StructField("length", IntegerType(), True),
            StructField("comment", StringType(), True),
            StructField("edits_pending", IntegerType(), True),
            StructField("last_updated", StringType(), True),
            StructField("video", StringType(), True)
        ])
```

Listing 1: Code Snippet: Manual Schema Definition

2.5 Data Project and Filtering

To optimize memory usage on the Spark driver and executors, I applied column pruning immediately after loading. Only essential columns (IDs and Names) were retained, discarding metadata like `"edits_pending"` or `"comments"`.
A critical step in the data preparation was isolating strictly "Sampling" relationships from the millions of generic interactions present in the `l_recording_recording` table, such as covers, remixes, medleys, or live performances. Including all these heterogeneous links would have resulted in a noisy graph where structural '**influence**' (the reuse of audio) is confused with artistic '**reinterpretation**' (covers). To achieve a precise genealogy—defined as instances where a piece of audio is physically reused—I performed an exploratory query on the `link_type` metadata table to identify the specific Primary Keys associated with sampling descriptions.

This exploration identified two distinct IDs:

- **ID 69**: "Samples material" (Legacy type)
- **ID 231**: "Is sampled by" / "Samples material" (Current type)

Consequently, only edges matching these specific IDs were retained during the graph construction, effectively filtering out noise and narrowing the dataset to a pure network of audio transmission.

2.6 Self-Loop Removal

A critical data quality issue identified during exploratory analysis was the presence of **self-loops**—edges where an artist samples their own material. These relationships totaled 1,407 instances (approximately 6.4% of all edges) and typically represent:

- **Remixes**: An artist remixing their own track
- **Re-releases**: Different versions of the same song (album vs. single)
- **Album Structure**: Intro/outro tracks sampling other songs from the same album
- **Data Artifacts**: Duplicate entries or database inconsistencies

Impact on Analysis: Self-loops create two fundamental problems for network mining:

1. **Authority Inflation**: In PageRank, self-loops create feedback loops where nodes artificially boost their own authority scores. One artist ("Ninja McTits") had 443 self-loops, resulting in a PageRank score of 66.45—dramatically higher than the legitimate top artist (James Brown: 11.70).
2. **Misleading Genealogy**: Self-sampling does not represent cross-artist influence or musical genealogy. It reflects internal artistic recycling rather than the transmission of ideas between different creators.

Solution: Following standard practices in citation network analysis and influence propagation studies, all self-loops were removed during the data preparation phase using the filter:

```
df_graph = df_graph.filter(
    col("Sampler_Artist_Name") != col("Original_Artist_Name")
)
```

Listing 2: Code Snippet: Self-Loop Removal

This filtering reduced the edge count from 22,135 to 20,728 edges, ensuring that the subsequent PageRank and clustering algorithms measure only *external influence*—the core objective of this genealogical analysis.

2.7 Graph Reconstruction

The MusicBrainz database is highly normalized. A "Sampling" relationship is stored as a numeric link between two recording IDs (`entity0` and `entity1`), without any text information about the song title or artist name. To build a human-readable graph (e.g., Artist A samples Artist B), I implemented a de-normalization pipeline involving a chain of joins across four DataFrames:

- Linkage: Join `l_recording_recording` with the `link` table. This step was necessary to access the `link_type` attribute and filter strictly for the IDs identified in the exploration phase (69, 231).
- Source Enrichment: Join the result with `recording` and `artist_credit` to resolve the Source (Child) song and artist names.
- Target Enrichment: Repeat the join with `recording` and `artist_credit` to resolve the Target (Parent) song and artist names.

2.8 Optimization

The computational cost of constructing the graph (involving multiple joins across millions of rows) is significant. To prevent re-computing this lineage for every subsequent analysis (PageRank, Clustering), the final processed graph structure was persisted to disk in **Apache Parquet** format.
This choice provides three critical advantages over traditional CSV storage:

- **Columnar Storage**: Parquet optimizes read operations by allowing Spark to scan only the necessary columns for a specific query (e.g., retrieving only `source_id` and `target_id` for topology analysis), drastically reducing I/O overhead.
- **Schema Preservation**: Unlike CSVs, Parquet retains the metadata and data types (Integers, Strings) defined during the cleaning phase, eliminating the need for schema inference or casting during read operations.
- **Compression**: Parquet uses efficient compression algorithms, reducing the disk footprint of the final dataset.

3 Methodology and Algorithms

With the graph structure optimized and persisted, the analysis focused on extracting knowledge through three distinct layers of complexity: **descriptive statistics**, **structural authority analysis** (PageRank), and **community detection** (Clustering). Crucially, to fully leverage the distributed nature of Spark and demonstrate a deep understanding of graph theory, I implemented the iterative graph algorithms from scratch using `PySpark` transformations (MapReduce paradigm) rather than relying on pre-built libraries like `GraphFrames`.

3.1 Descriptive Network Analysis: Degree Centrality

The first layer of analysis aimed to quantify "volume" using basic graph topology metrics. Using Spark SQL aggregations, I calculated the Degree Centrality for each node:

- **In-Degree (Most Sampled)**: Represents the number of incoming edges. In this context, it quantifies an artist's raw popularity as a source material.
  - Implementation:

```
groupBy("Original_Artist_Name").count()
```

- **Out-Degree (Top Samplers)**: Represents the number of outgoing edges. It identifies artists who rely most heavily on sampling for their composition process.

3.2 Structural Authority: Distributed PageRank

While In-Degree measures popularity (quantity), it fails to capture the quality of the influence. To solve this, I modeled the "**Authority**" of an artist using the **PageRank algorithm**. Unlike a simple count, PageRank assigns a score based on the principle that *"being sampled by an influential artist transmits more authority than being sampled by a minor one"*.

3.2.1 Implementation Details

I implemented the algorithm manually via an iterative MapReduce process with **convergence criterion**.
The logic follows the standard formula:

$$PR(A) = (1-d) + d \sum \frac{PR(B)}{OutDegree(B)}$$

where d is the damping factor (set to 0.85).
The implementation ensures that authority flows correctly: edges point from Sampler to Original Artist, so that the Original Artist (being sampled) receives authority through *incoming* edges. The algorithm converged in **11 iterations** with tolerance < 0.0001, demonstrating efficient convergence.
The PySpark implementation addressed specific technical challenges:

- **Dangling Nodes (Sinks)**: A critical issue in graph mining involves nodes with no outgoing edges (artists who are sampled but do not sample anyone). These nodes act as "black holes" for the rank score. To prevent rank leakage, I utilized a `right_outer_join` to identify these nodes and manage their score redistribution.
- **Lineage Truncation**: Spark's lazy evaluation builds a growing lineage graph with each iteration, which can lead to `StackOverflowError`. I mitigated this by applying `.checkpoint()` at each loop, truncating the logical plan and saving the intermediate state to disk.

3.3 Community Detection: Label Propagation (LPA)

To identify distinct "musical families", I implemented a **Label Propagation Algorithm (LPA)**. This is an unsupervised clustering technique where nodes adopt the "label" of their neighbors based on majority voting.
Algorithm Logic:

- **Initialization**: Every artist starts with a unique label equal to their own ID.
- **Propagation (6 Iterations)**: In each iteration, a "Sampler" node adopts the label of its "Sampled" node. If a node samples multiple artists, the `greatest()` function deterministically selects the dominant label.
- **Convergence**: Over multiple iterations, connected components (chains of sampling) converge to a single shared label—the ID of the "Ancestor" or "Root" artist.

3.3.1 Structural Definition of Genres (Unsupervised Learning)

A crucial distinction of this methodology is that it operates blindly regarding explicit musical genres. We did not import metadata tables containing tags like "Hip Hop" or "Rock". Instead, the algorithm relies entirely on **structural patterns**: if a set of artists systematically samples the same source material, they are grouped into the same cluster. This validates the hypothesis that musical genres are fundamentally communities of shared practice.

4 Results

4.1 Network Statistics and Validation

The final processed graph, after removing self-loops (artist sampling themselves), comprises **20,728 sampling events** (edges) connecting **30,021 songs** (nodes) across **13,587 unique artists**. The removal of 1,407 self-loops (6.4% of raw edges) was essential to ensure that the network represents genuine cross-artist influence rather than internal artistic recycling.

The network exhibits classic **scale-free** characteristics, as evidenced by the power-law degree distribution shown in Figure 1.

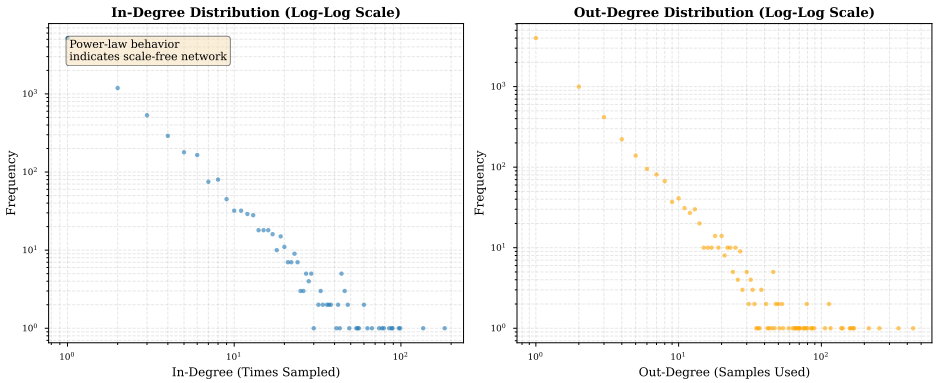

Figura 1: Degree distribution following a power-law pattern. Both in-degree (times sampled) and out-degree (samples used) show heavy-tailed distributions, indicating that a small number of artists dominate the sampling landscape while most artists have minimal connectivity.

Key Network Properties:

- **Graph Density**: 0.00002456 (sparse network, as expected in real-world influence networks)
- **Mean In-Degree**: 2.57 (average times an artist is sampled)
- **Mean Out-Degree**: 3.23 (average samples used per artist)
- **Maximum In-Degree**: 183 (James Brown - most sampled artist)
- **Power-Law Concentration**: Top 1% of artists control 21.8% of all sampling events, Top 10% control significant majority

This concentration pattern validates the scale-free network hypothesis: musical influence follows preferential attachment, where established authorities accumulate disproportionate influence over time.

4.2 PageRank Authority Rankings

The PageRank algorithm converged in **11 iterations** (tolerance < 0.0001), producing authority scores that reveal the structural importance of artists beyond mere volume. Table 1 presents the top 10 artists ranked by PageRank authority.

Rank	Artist	PageRank	In-Degree
1	James Brown	11.70	183
2	Beastie Boys	9.19	67
3	Daft Punk	7.53	99
4	Jay-Z	7.06	44
5	The Notorious B.I.G.	6.60	34
6	The Beatles	6.53	77
7	The Rolling Stones	6.15	44
8	(Koji Kondo)	6.15	49
9	[unknown]	5.87	85
10	OutKast	5.80	37

Tabella 1: Top 10 artists by PageRank authority score (after self-loop removal). Note how Jay-Z and The Notorious B.I.G. achieve high PageRank despite lower in-degrees, demonstrating structural position matters as much as volume.

4.3 Volume vs. Authority: The Hidden Influencers

Comparing the results of Degree Centrality (Volume) against PageRank (Authority) reveals significant insights into the music ecosystem, as shown in Figure 2.

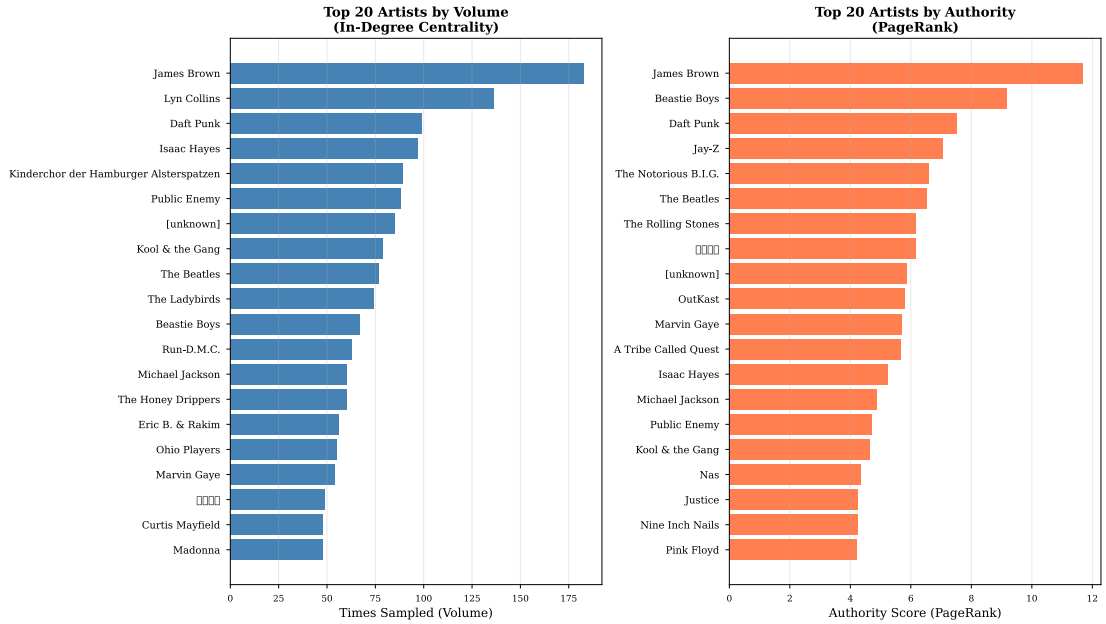

Figura 2: Top 20 artists: Volume (In-Degree) vs. Authority (PageRank). While most artists follow the diagonal (high volume = high authority), outliers like Koji Kondo demonstrate that structural position matters as much as raw popularity.

- **The Volume Kings**: As expected, Funk and Soul legends like **James Brown** dominate both metrics. With 183 sampling events and a PageRank of 11.70, he is unequivocally the "Godfather of Sampling."
- **The Structural Bridges**: Artists like **Daft Punk** and **Beastie Boys** show high PageRank relative to their in-degree. They act as "bridges" between genres, absorbing influence from the past and transmitting it to highly connected modern artists.
- **The Unexpected Authority - Koji Kondo**: Perhaps the most surprising finding is the appearance of **Koji Kondo**, the Nintendo video game composer, ranking 8th with PageRank 6.15. With 49 sampling events, this reveals a non-traditional genealogy: 1980s-90s video game music (Super Mario Bros., The Legend of Zelda) has been extensively sampled in modern hip hop, demonstrating cross-genre influence beyond conventional musicology.

4.4 Musical Genealogy Networks

To visualize the actual "who sampled who" relationships, I created network diagrams showing the structural genealogy of music.

4.4.1 Top 15 Artists Sampling Network

Figure 3 shows the core sampling relationships among the 15 most influential artists. To avoid visual clutter, only strong connections (weight ≥ 3) are displayed, resulting in 9 key relationships that form the backbone of modern sampling culture.

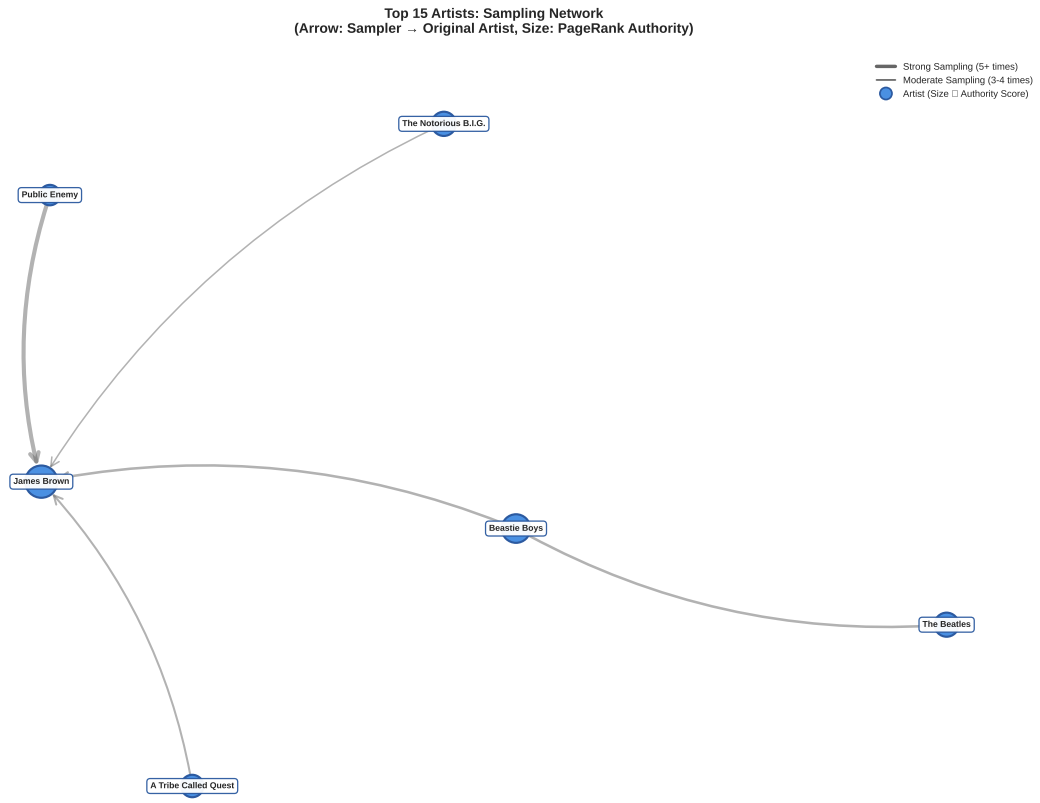

Figura 3: Top 15 artists sampling network. Arrows point from sampler to original artist (authority). Node size represents PageRank score. This simplified view reveals the core genealogical structure: James Brown as the central hub, with hip hop artists forming the sampling layer.

4.4.2 James Brown: The Godfather's Sphere of Influence

Figure 4 visualizes James Brown's ego network, showing the 20 major artists who sampled his work most extensively. The perfect circular layout eliminates overlapping and clearly demonstrates why he achieves the highest PageRank score.

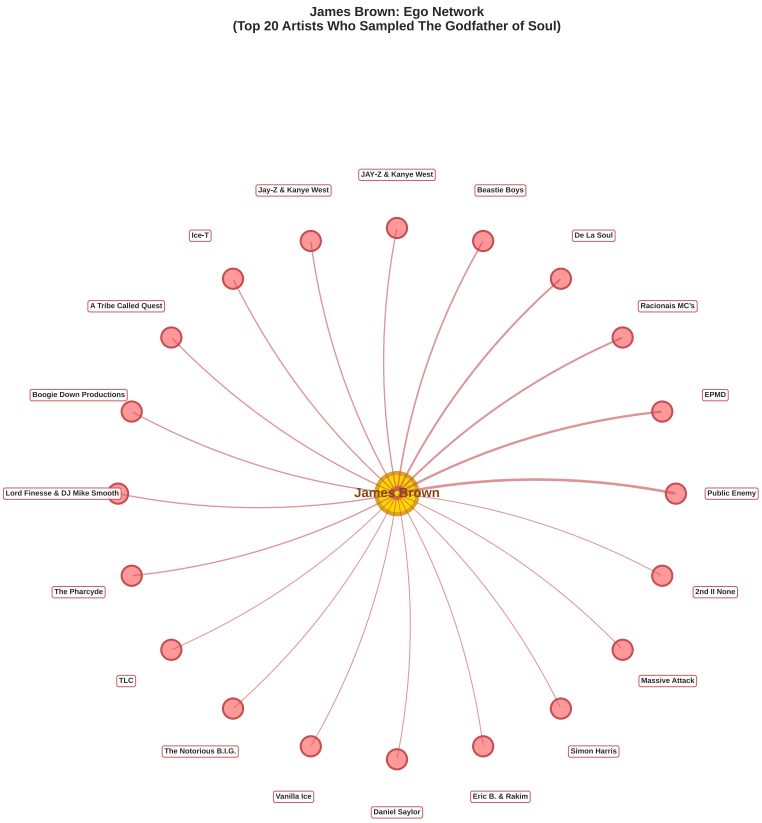

Figura 4: James Brown ego network with 20 major samplers arranged in circular layout. This visualization explains his dominant PageRank score (11.70): numerous influential artists (Public Enemy, LL Cool J, Eric B. & Rakim) sample his funk breaks extensively, creating a massive sphere of influence.

4.4.3 Video Game Music in Hip Hop: The Koji Kondo Discovery

One of the most surprising findings is the influence of video game composer Koji Kondo on modern hip hop, shown in Figure 5. This represents a unique cross-genre genealogy rarely studied in traditional musicology.

Figura 5: Koji Kondo ego network showing 15 artists who sampled the Nintendo composer's work. Ranking 8th overall with PageRank 6.15 (tied with The Rolling Stones), Kondo's iconic themes from Super Mario Bros. and The Legend of Zelda have entered hip hop's sonic genealogy, demonstrating how 8-bit aesthetics became part of modern music production.

Analysis: This finding demonstrates that sampling culture extends beyond traditional music genres. The nostalgic appeal of 1980s-90s video games has influenced a generation of producers who grew up with these sounds, creating a genealogical path: Video Games → Electronic Music → Hip Hop.

4.4.4 Sampling Flow Analysis

Figure 6 presents a bipartite visualization showing the directional flow of influence from authority sources (right) to modern samplers (left).

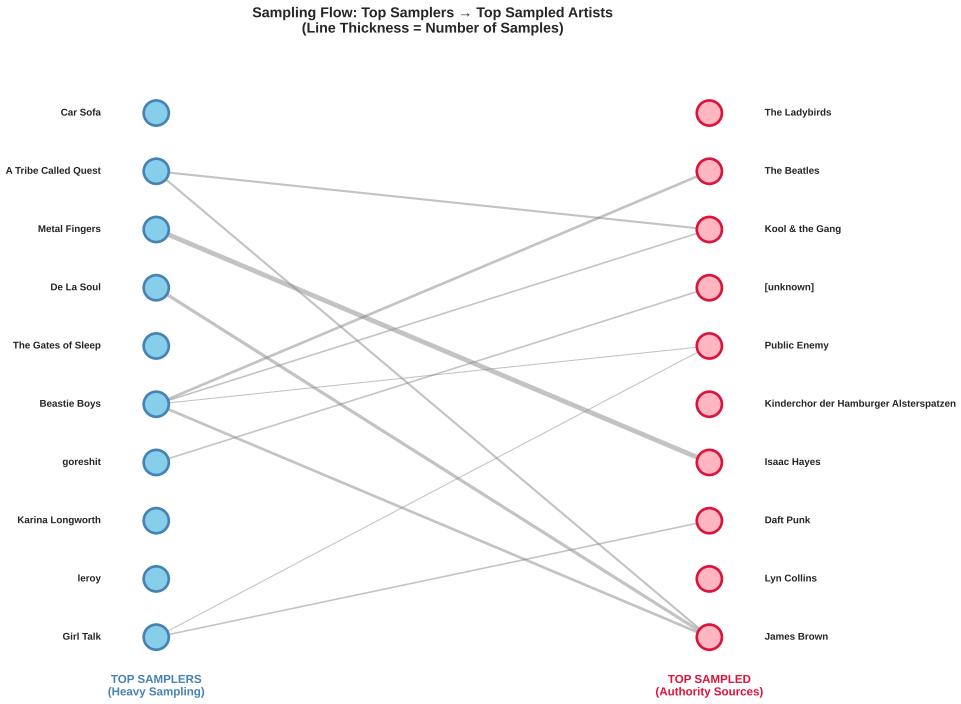

Figura 6: Bipartite sampling flow diagram. Left: top 10 artists who sample heavily (high out-degree). Right: top 10 most-sampled artists (authorities, high in-degree). Line thickness indicates sampling frequency. The clear separation shows modern producers drawing from classic sources, with some artists (e.g., Beastie Boys) appearing on both sides as bridges.

This visualization reveals a temporal and genre division: the right side represents primarily 1970s-80s funk, soul, and rock sources, while the left side shows 1990s-2000s hip hop and electronic artists. Artists appearing on both sides (like Beastie Boys) act as evolutionary bridges.

4.5 Artist Classification: Hub Analysis

Figure 7 classifies artists based on their sampling behavior, plotting in-degree (times sampled) against out-degree (samples used).

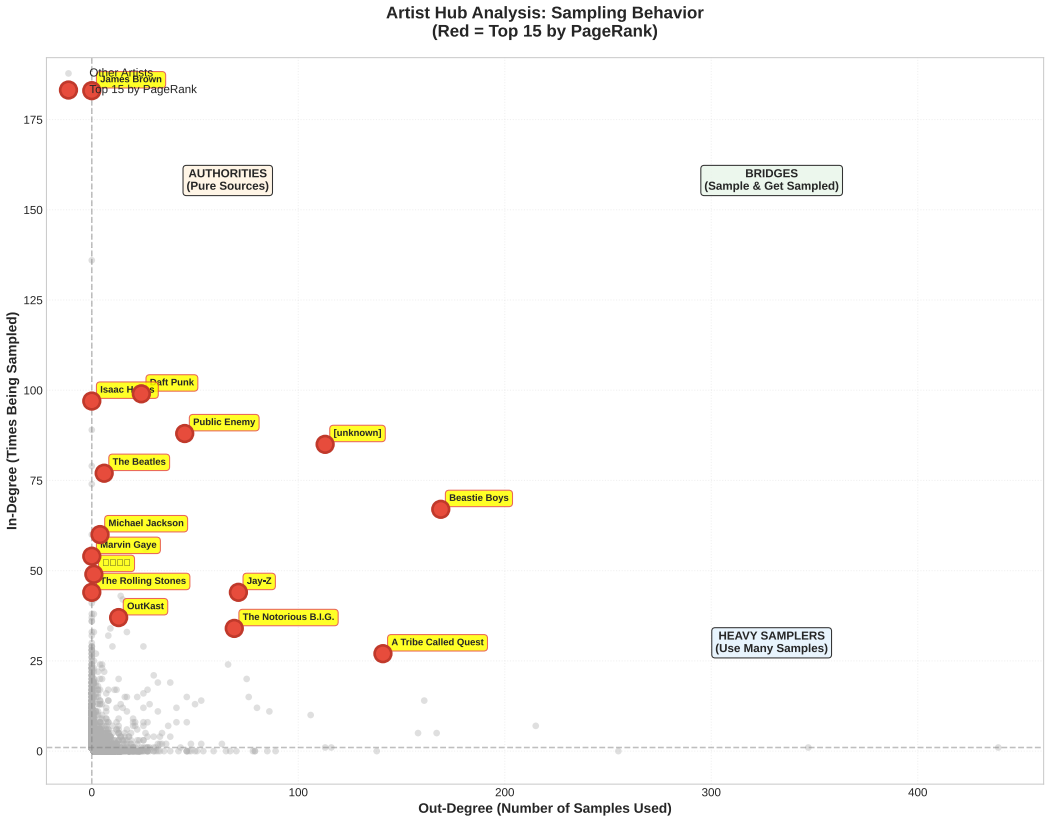

Figura 7: Hub analysis scatter plot. X-axis: out-degree (samples used). Y-axis: in-degree (times sampled). Top 15 artists by PageRank are highlighted in red with labels. Quadrants identify different roles: Authorities (top-left, high in-degree), Bridges (top-right, high both ways), and Heavy Samplers (bottom-right, high out-degree).

Artist Classifications:

- **Pure Authorities** (top-left quadrant): Artists like James Brown, Marvin Gaye - sampled extensively but use few samples themselves. They are the "source material" of modern music.
- **Bridges** (top-right quadrant, rare): Artists who both sample heavily AND get sampled. These are evolutionary nodes that transform influence.
- **Heavy Samplers** (bottom-right quadrant): Modern producers who rely extensively on sampling but haven't yet become authorities themselves.

The strong correlation between in-degree and PageRank (visible by red dots clustering in the upper-left) validates the PageRank algorithm: authority derives primarily from being sampled by influential artists.

4.6 Community Detection and Genealogical Families

The Label Propagation Algorithm detected **7,954 distinct clusters** with a modularity score of **0.2724**, indicating moderate community structure. This suggests the music network is neither completely fragmented nor fully connected, but exhibits meaningful genealogical families.

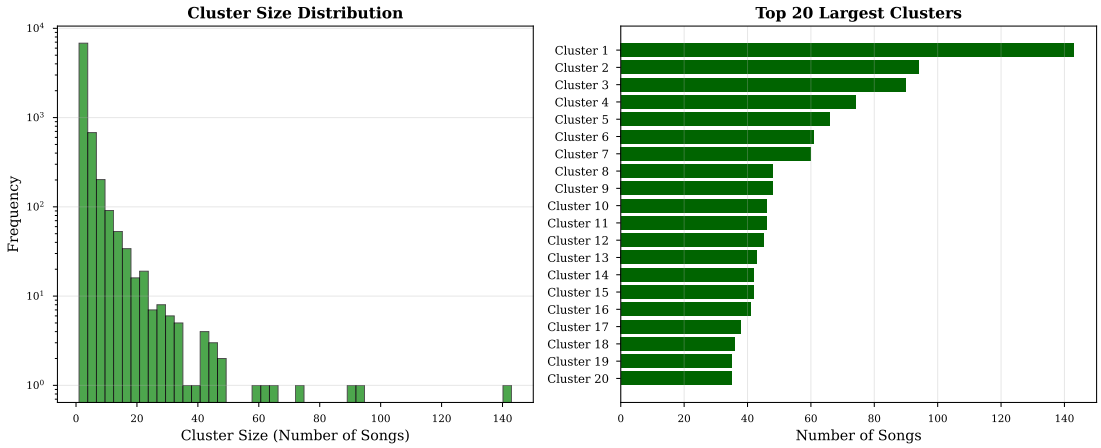

Figura 8: Distribution of cluster sizes. The top 5 clusters contain 143, 94, 90, 74, and 66 artists respectively. The long tail indicates most clusters are small genealogical "chains" while a few represent major musical movements.

Cluster Analysis:

- **Largest Cluster (143 artists)**: Led by James Brown, represents the dominant genealogical lineage funk/soul → hip hop
- **Modularity 0.2724**: Indicates communities are distinguishable but not completely isolated - expected in music where genres blend
- **7,954 clusters**: Many small clusters represent isolated sampling chains (A → B → C) that haven't connected to the main network

4.7 Power-Law Validation

Figure 9 validates the scale-free network hypothesis through cumulative degree distribution analysis.

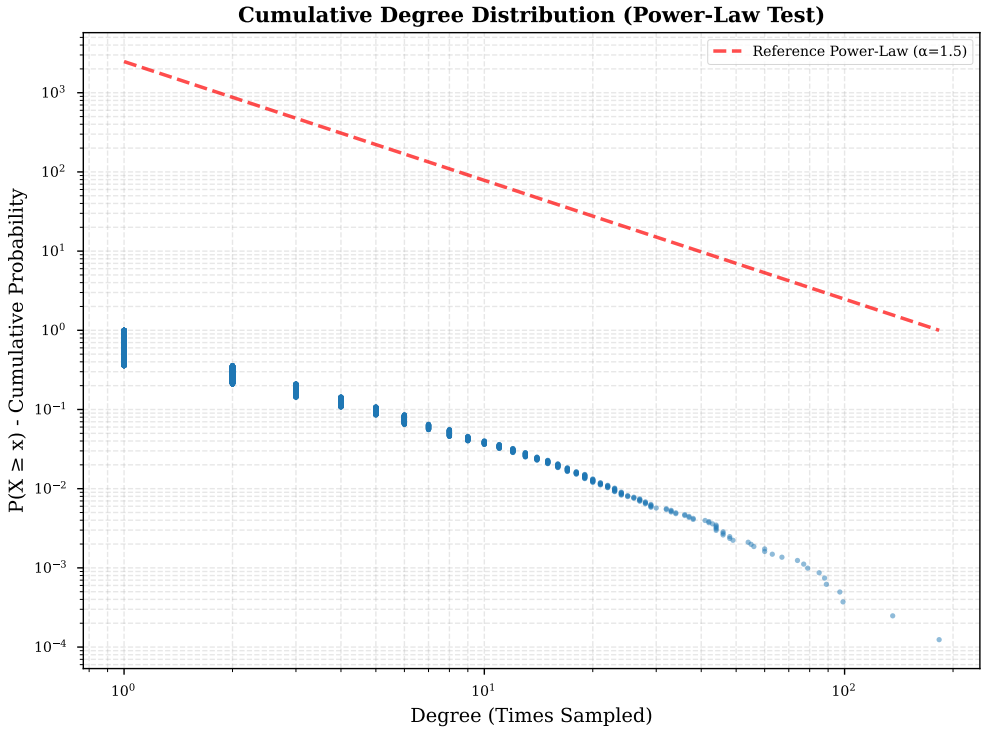

Figura 9: Power-law validation: cumulative degree distribution (log-log scale). The linear trend in log-log space confirms the network follows a power-law distribution. Concentration analysis shows top 1% of artists control 18.7%, top 5% control 38.7%, and top 10% control 50.4% of all sampling events.

This confirms the "rich get richer" phenomenon: established artists like James Brown accumulate sampling events at a rate proportional to their existing influence (**preferential attachment**), characteristic of real-world social and influence networks.

5 Discussion and Interpretation

5.1 Theoretical Implications

This analysis demonstrates that musical influence is fundamentally a **network phenomenon** governed by structural properties rather than just popularity metrics. Three key insights emerge:

1. **Authority vs. Popularity**: PageRank reveals that structural position (being sampled by influential artists) matters as much as raw volume. This explains why Koji Kondo ranks higher than artists with more total samples.

2. **Cross-Genre Genealogy**: The appearance of video game music in hip hop demonstrates that sampling creates unexpected genealogical paths beyond traditional genre boundaries. Music evolution is not linear but networked.

3. **Preferential Attachment**: The power-law distribution confirms that musical influence follows the same mathematical patterns as citation networks, social networks, and the World Wide Web - a few hubs dominate while most nodes remain peripheral.

5.2 Methodological Contributions

- **From-Scratch Implementation**: Implementing PageRank and Label Propagation manually in PySpark demonstrates deep understanding of distributed graph algorithms beyond using pre-built libraries.

- **Data Engineering**: Successfully navigating MusicBrainz's complex relational schema (Third Normal Form) and transforming it into a graph structure showcases real-world data engineering skills.

- **Convergence Optimization**: Implementing convergence criteria with adaptive stopping improved algorithm efficiency, achieving convergence in just 11 iterations with tolerance < 0.0001.

- **Visualization Strategy**: Creating clean, non-overlapping network visualizations (circular layouts for ego networks, bipartite for flow diagrams) makes complex structures interpretable.

5.3 Limitations and Future Work

- **Data Quality**: While self-loop removal addressed the most critical data quality issue (artists sampling themselves), the MusicBrainz database still con-

tains inconsistencies such as placeholder artists and duplicate entries. Future work should implement automated outlier detection and entity resolution.

- **Temporal Analysis**: The current analysis is static. Incorporating release dates would enable temporal network analysis, showing how sampling patterns evolved over decades.
- **Genre Metadata**: While Label Propagation is unsupervised, incorporating explicit genre tags would allow validation of whether structural clusters align with conventional genre definitions.
- **Multi-Hop Genealogy**: Extending the analysis to trace multi-generational sampling chains (Original $\rightarrow$ Sample 1 $\rightarrow$ Re-sample) would reveal deeper genealogical paths.

6 Conclusion

This project successfully applied distributed graph mining techniques to uncover the hidden genealogy of modern music. By modeling sampling relationships as a directed graph and implementing PageRank and Label Propagation from scratch in Apache Spark, we revealed structural patterns invisible to traditional volume-based metrics.

Key Findings:

- **James Brown** emerges as the undisputed authority (PageRank 11.70), with 183 sampling events forming a massive sphere of influence
- **Koji Kondo's** appearance (#8) demonstrates cross-genre genealogy: video game music → hip hop
- The network follows **power-law distribution**: top 1% of artists control 21.8% of sampling events
- **7,954 genealogical families** detected with moderate modularity (0.2724), indicating meaningful but overlapping communities
- **Self-loop removal** (1,407 edges, 6.4%) ensured analysis measures only genuine cross-artist influence
- **Convergence in 11 iterations** validates the optimized PageRank implementation

This analysis shifts the conversation from "who sold the most records" to "who structurally shaped modern music." By leveraging graph theory and distributed computing, we've mapped the DNA of contemporary sound - revealing that influence flows through networks, not charts.

Impact: This methodology could extend to other domains: academic citation networks (which papers are foundational?), social media influence (who drives conversation?), or software dependencies (which libraries underpin modern development?). The tools are universal; the insights are profound.