

Analytical Workflow: Community Structure & Evolution in the IEEE VIS Co-Authorship Network

Pipeline overview

plaintext

```
Raw records  Time-windowed co-authorship networks  Per-window community  
Cross-window matching (lineages)  Core/leader analysis  Topic character  
Cross-community overlap analysis  Coordinated visual front-end  Domain-
```

Because the dataset is small enough (3,596 papers) for fully interactive analysis, the workflow is designed as an **iterative, human-in-the-loop visual analytics pipeline**: every algorithmic stage produces an intermediate artefact that is inspected visually before the next stage runs, and parameters can be revised based on domain knowledge.

1. Network construction

- Build a paper–author table (year, sub-conference Vis/InfoVis/VAST/SciVis, title, abstract, keywords, citations, disambiguated author ID).
- Construct **overlapping sliding time windows** (e.g., 5-year windows, 1-year step) rather than cumulative or single-year graphs — this smooths sparse early years while keeping enough temporal resolution to detect merges/splits.
- Edge weight = co-authored papers in the window (optionally fractional-count normalised for author-list length).

Check (visual): density/#components/giant-component-size over time, plus force-directed layout of a few sample windows — sanity check before detection.

2. Per-window community detection

- Use **Leiden** with the **CPM objective** (not plain modularity — the field's size grows ~10× from 1990–2024, so modularity's resolution limit would bias detection toward large late-period, small early-period communities).
- Run multiple seeds consensus clustering to get robust partitions; expose resolution parameter γ interactively.

Check (visual): resolution profile (γ vs. $\#$ communities/modularity); node-link/matrix view colored by community; cross-tab heatmap community $\times$ sub-conference as a domain-knowledge sanity check.

3. Cross-window matching community lineages

- Similarity between communities in adjacent windows = Jaccard overlap of author sets.
- Classify transitions with standard rules: **birth, death, continuation, growth/shrink, merge (n1), split (1n)**.
- Assign persistent **lineage IDs** to tracked communities.

Check (visual): alluvial/storyline diagram (node size = community size, ribbon width = shared members) — the primary artefact for answering "how many communities, do they merge/split/dissolve, when". Analyst annotates events using domain knowledge (e.g., the 2013 unification of InfoVis/VAST/SciVis into one VIS proceedings is a plausible structural break to check against).

4. Core / leadership / stability analysis

- Per lineage, build an author $\times$ window presence matrix.
- **Persistence score** = fraction of windows in the lineage's span where the author is present.
- **Centrality** (degree/eigenvector, citation-weighted) per window.
- **Core** = high persistence + high centrality; **leaders** = top core members by citations/centrality.
- **Core stability index** = mean pairwise Jaccard of the core set across consecutive windows.

Check (visual): heatmap (authors $\times$ windows, sorted by persistence), small multiples across lineages; ranked leader table.

5. Cross-community overlap (bridging authors)

- Per window, compute each author's **participation coefficient** across communities (bridges vs. loyal members).
- Aggregate an inter-community network (meta-nodes = communities, edge weight = shared/collaborating authors).

Check (visual): aggregated meta-network (animated/slider over time); **UpSet plot** of author-set memberships across lineages.

6. Topic characterisation

- For papers attached to each community-window node: TF-IDF top terms or topic model (LDA/BERTopic) on title+abstract+keywords.
- Track topic profile along each lineage.

Check (visual): word clouds/top-term lists attached to storyline nodes; **ThemeRiver** of topic proportions per lineage; UMAP projection of community-window topic vectors colored by lineage (reveals topical convergence before merges / divergence before splits).

7. Integrated visual front-end (coordinated views)

Linked views with brushing: storyline (evolution) network snapshot core heatmap topic panel citation-impact overlay (bubble size/line chart). Selecting a lineage or window highlights consistently across all views; hovering a node shows top authors, keywords, and papers.

8. Validation & iteration

- Compare detected leaders/topics against known VIS researchers/subfields; check alignment with sub-conference labels and known historical events (e.g., conference restructuring).
- If misaligned, revisit window size, γ , or matching threshold; run sensitivity analysis (small multiples of storylines under varied parameters) to confirm robustness of major lineages.

Mapping questions methods visualisations

Analyst question	Method	Key visualisation
How many communities exist?	Leiden (CPM) per window + lineage tracking	Storyline/alluvial diagram, community-count timeline
Core stability & leaders?	Persistence $\times$ centrality on author–window matrix	Sorted heatmap, ranked leader table

Analyst question	Method	Key visualisation
Main research foci & evolution?	Topic modelling per community-window	ThemeRiver, word clouds on storyline nodes
Membership overlap between communities?	Participation coefficient, meta-network	Aggregated community graph, UpSet plot
Merges/splits/dissolutions — when/why?	Rule-based transition classification	Annotated alluvial diagram + topic/UMAP cross-check