Graph Mining

Thomas Bonald and Tiphaine Viard

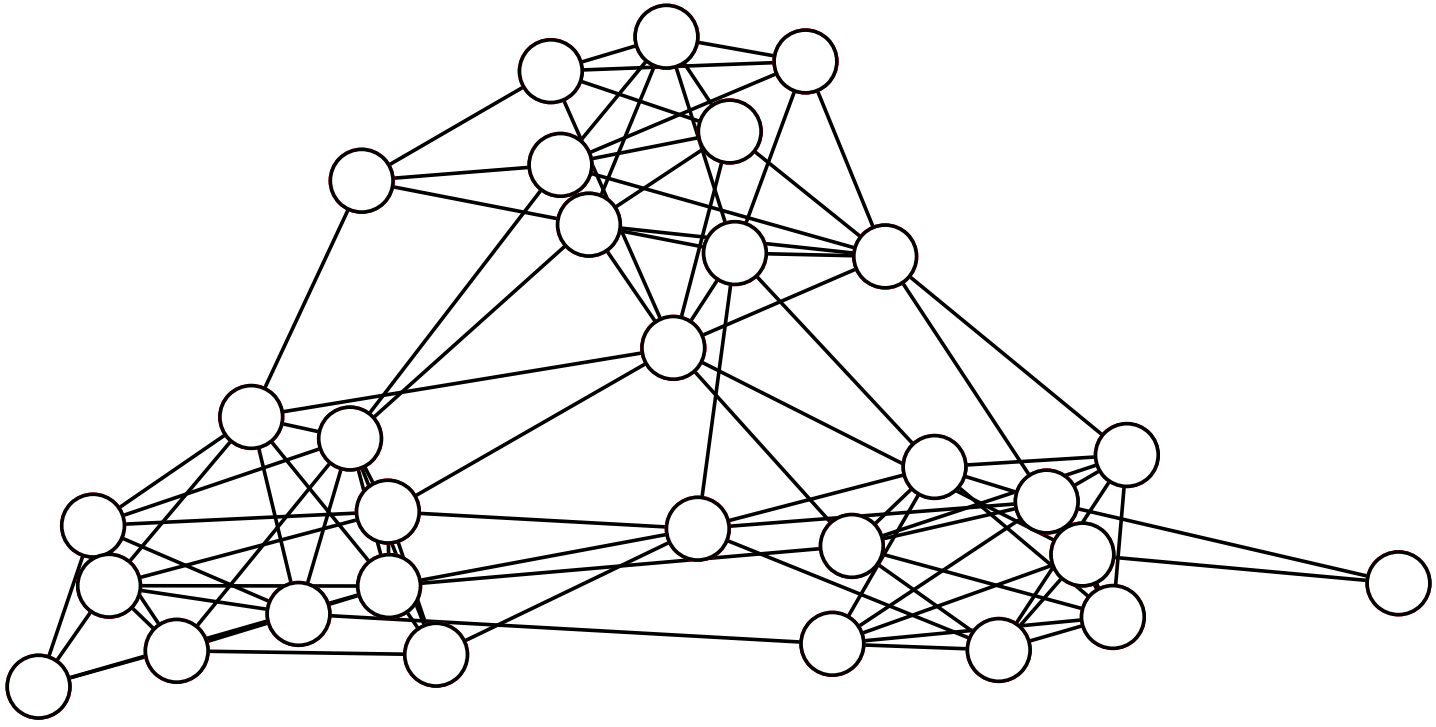
2020 – 2021

Outline

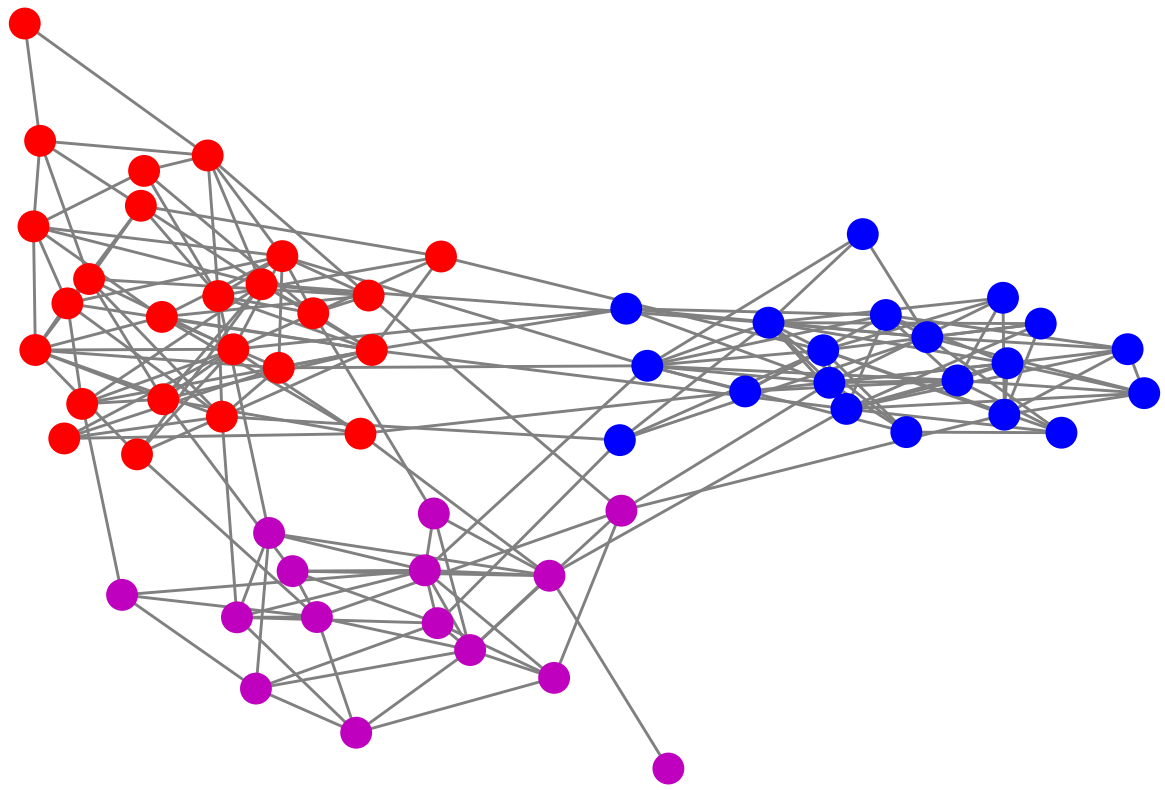
1. Sparse matrices
2. PageRank
3. **Clustering**
4. Embedding and neural networks

Graph clustering

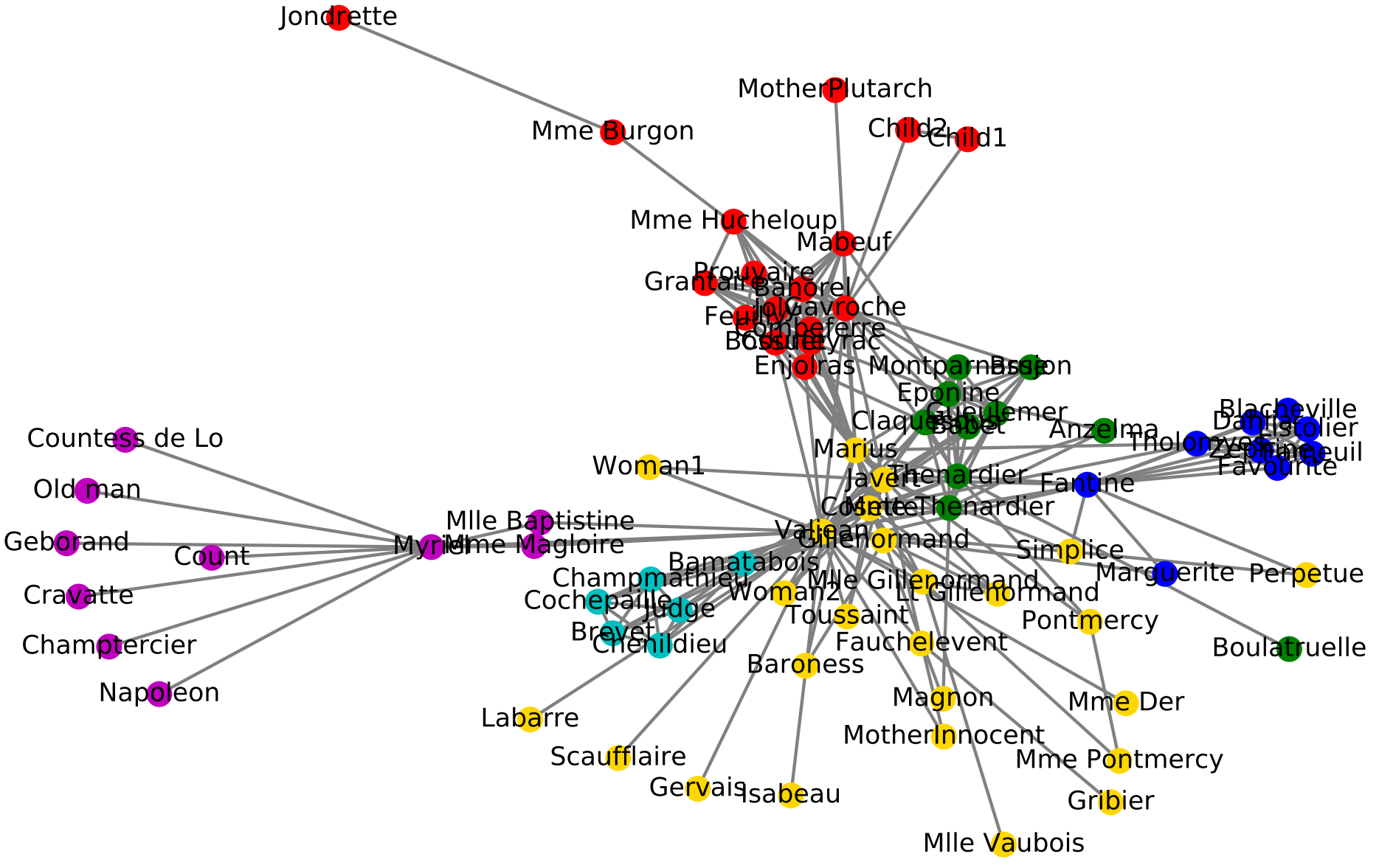
How to identify relevant groups of nodes in a graph?



Example

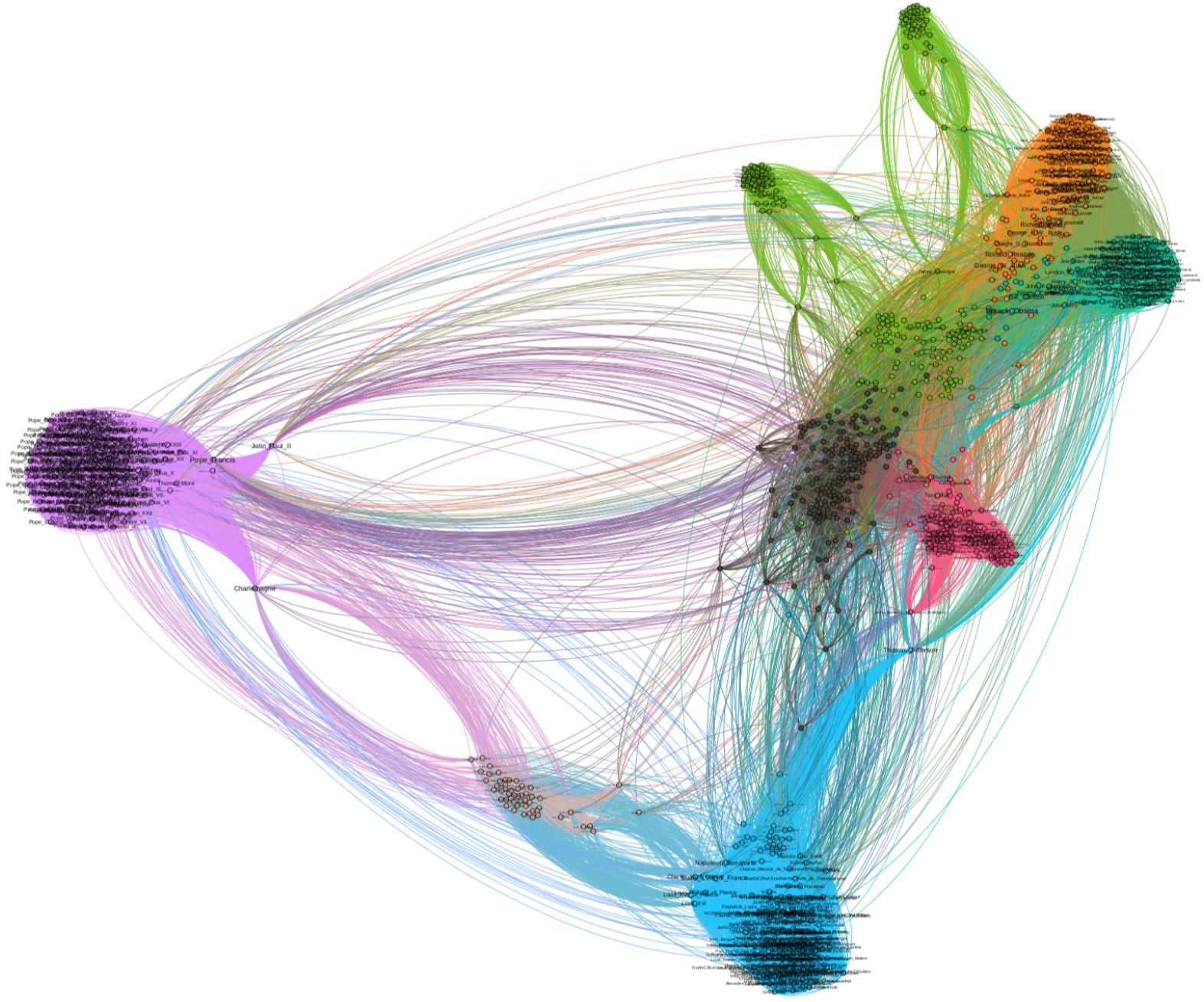


Characters of Les Miserables

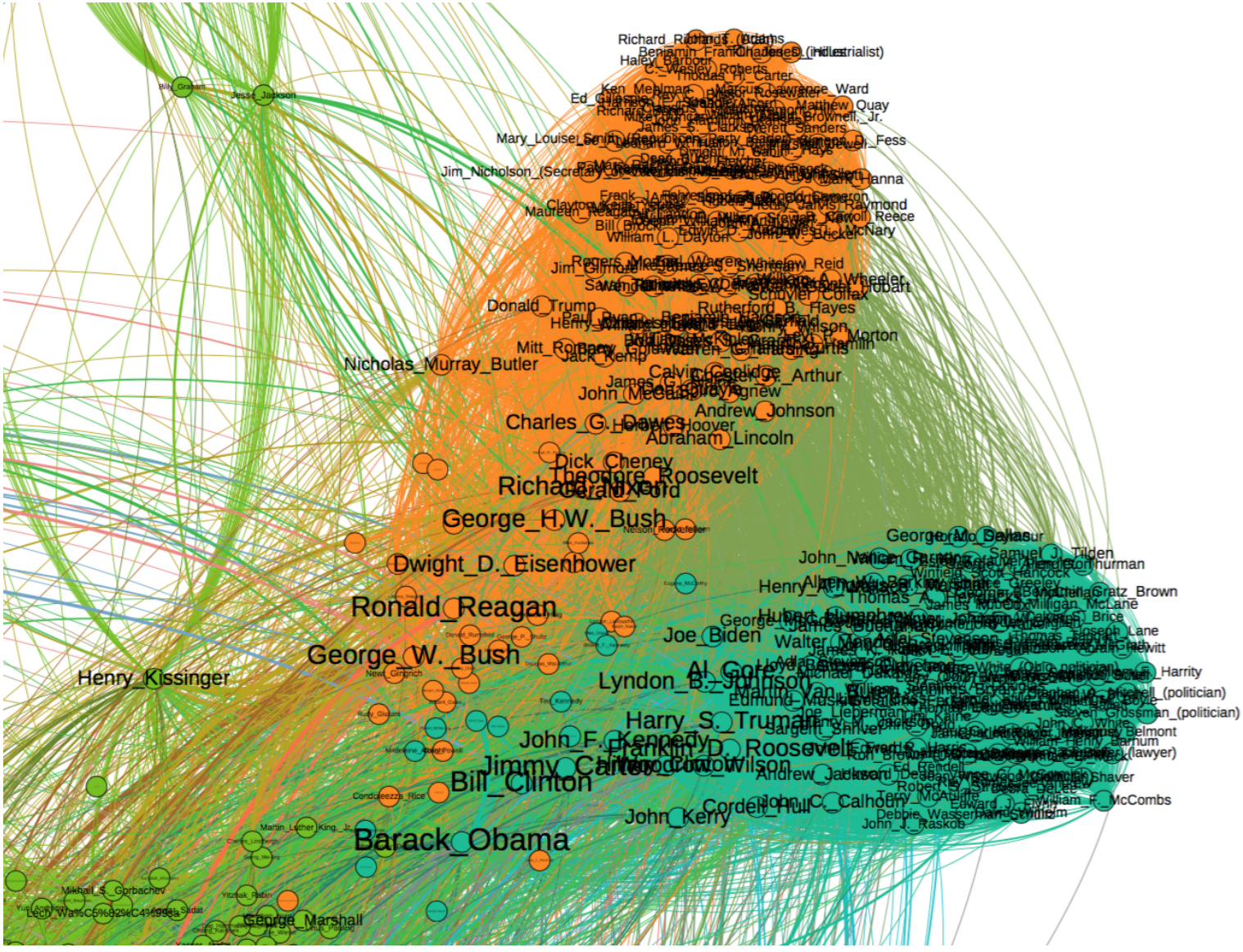


OpenFlights

Wikipedia restricted to Political Figures



Wikipedia restricted to Political Figures (zoom)



Graph clustering

- The clustering of a graph $G = (V, E)$ of n nodes and m edges is any function $C : V \to \{1, \ldots, K\}$

Graph clustering

- The clustering of a graph $G = (V, E)$ of n nodes and m edges is any function $C : V \to \{1, \ldots, K\}$
- In general, K is unknown (unlike K-means) and we look for the best clustering **irrespective** of the value of K

Graph clustering

- The clustering of a graph $G = (V, E)$ of n nodes and m edges is any function $C : V \to \{1, \ldots, K\}$
- In general, K is unknown (unlike K-means) and we look for the best clustering **irrespective** of the value of K
- We first assume that the graph is **undirected** and **unweighted**

Modularity

The modularity of clustering C is defined by:

$$Q(C) = \frac{1}{2m} \sum_{i,j \in V} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta_{C(i),C(j)}$$

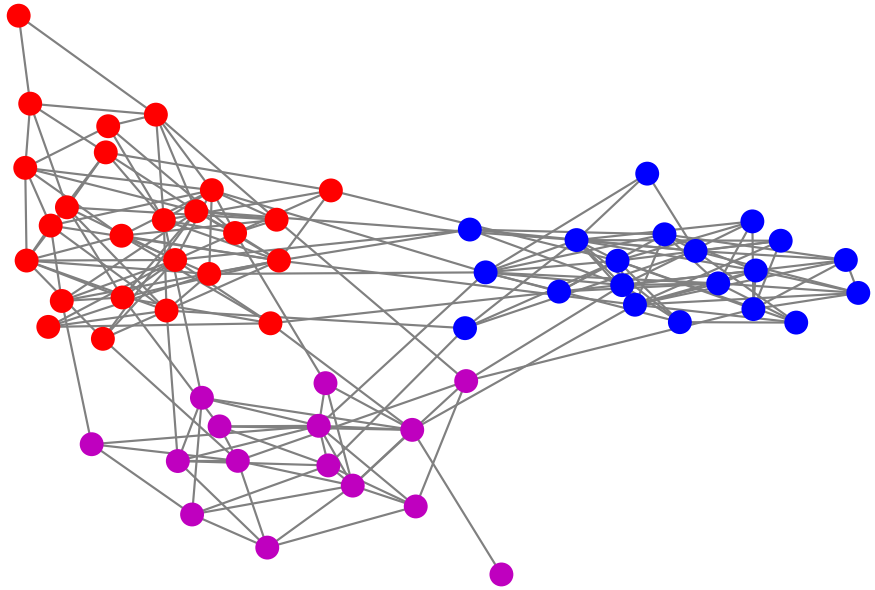
where m is the number of edges

Maximizing the modularity

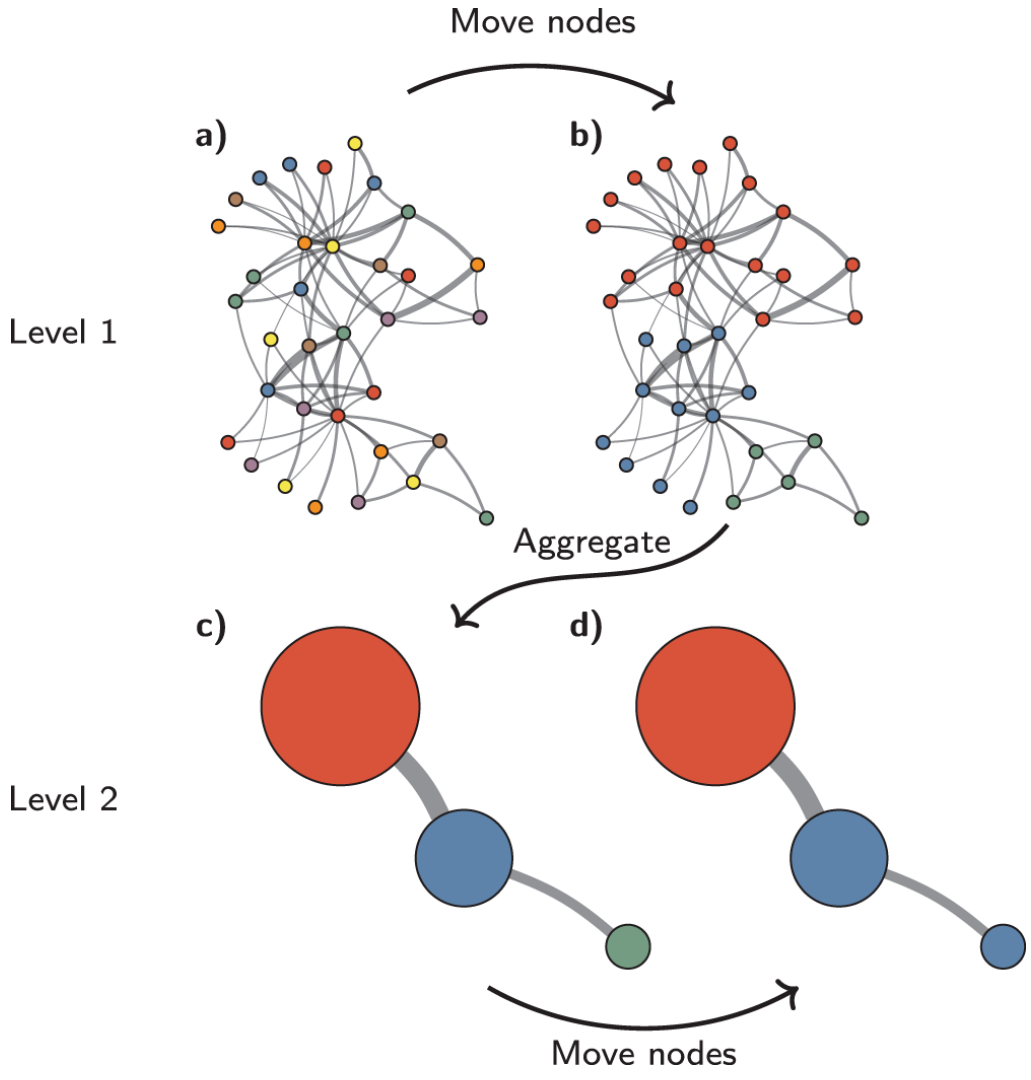
Consider the following problem:

$$\max_{C} Q(C)$$

- Combinatorial problem!
- NP-hard



The Louvain algorithm



(Traag et al., 2019)

The Louvain algorithm

Greedy algorithm:

1. **(Initialization)** $C \leftarrow$ identity
2. **(Maximization)** While modularity $Q(C)$ increases, update C by moving one node from one cluster to another
3. **(Aggregation)** Merge all nodes belonging to the same cluster into a single node, update the weights accordingly and apply step 2 to the aggregate graph

The Louvain algorithm

Greedy algorithm:

1. **(Initialization)** $C \leftarrow$ identity
2. **(Maximization)** While modularity $Q(C)$ increases, update C by moving one node from one cluster to another
3. **(Aggregation)** Merge all nodes belonging to the same cluster into a single node, update the weights accordingly and apply step 2 to the aggregate graph

Note: The outcome depends on the order in which nodes are considered!

Extensions

- Weighted graphs:

$$Q(C) = \frac{1}{w} \sum_{i,j \in V} \left(A_{ij} - \frac{w_i w_j}{w} \right) \delta_{C(i),C(j)}$$

- Resolution parameter:

$$Q_\gamma(C) = \frac{1}{w} \sum_{i,j \in V} \left(A_{ij} - \gamma \frac{w_i w_j}{w} \right) \delta_{C(i),C(j)}$$

- Directed graphs: seen as bipartite graphs, i.e.,

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

Bipartite modularity definition is an **open problem** (but you can do **co-ranking**)