Using Stable Communities for Modularity Maximization

Sanjukta Bhowmick and
Sriram Srinivasan
University of Nebraska at Omaha
Feb 13 2012

Introduction

- Networks are models of systems of interacting/ interdependent entities
  - Vertices ⇔ Entities, Edges ⇔ Relations
- Analysis Objective: Find communities—groups of vertices that have denser connections within groups and sparser connections across groups.
- Modularity: Q = *Σi (C(i,i)-a(i)^2)*
  - *Fraction of within community edges – expected edges for random connections*

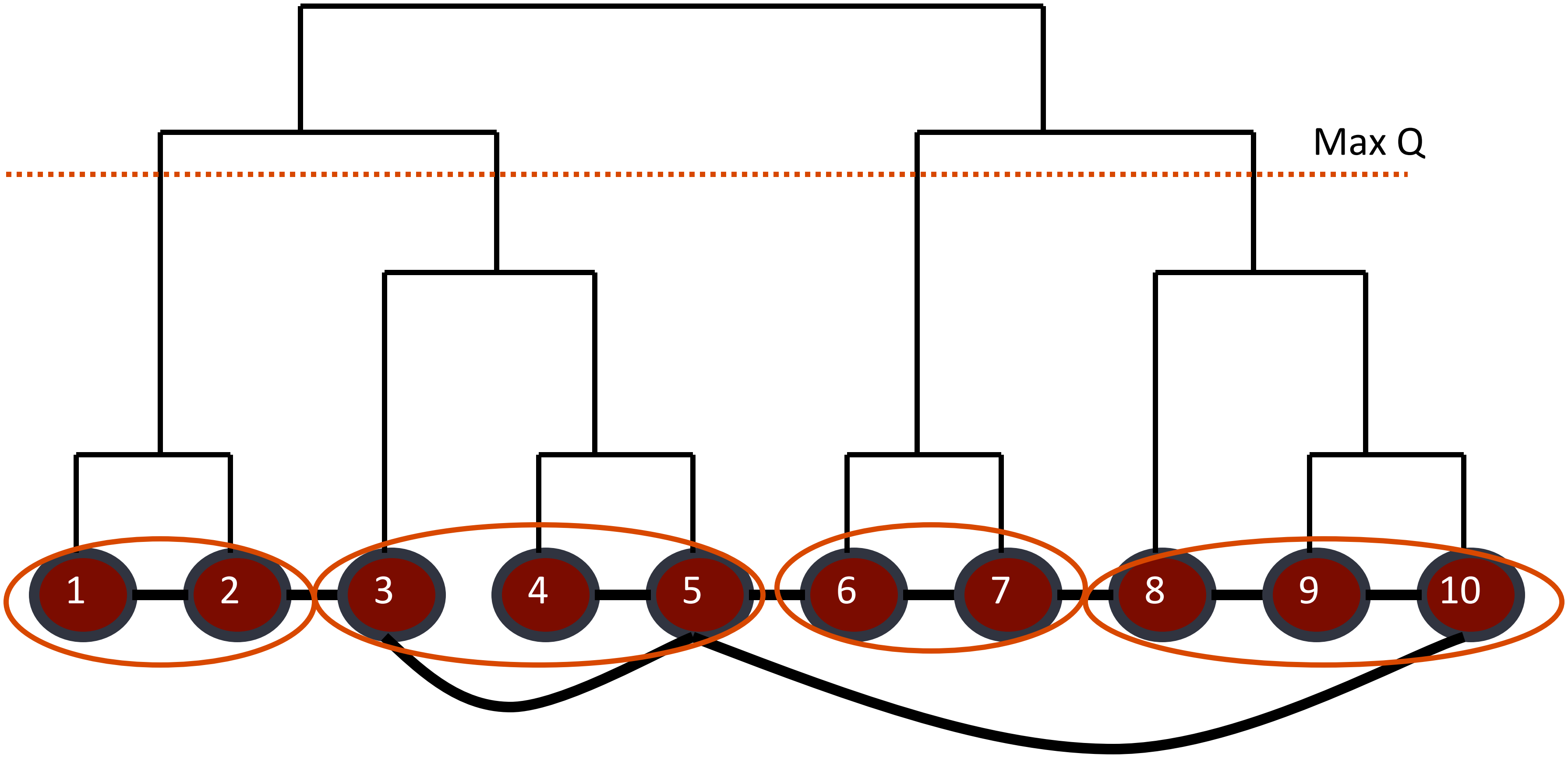
Modularity Maximization

- Higher modularity indicates better community –"*height of a peak is a measure of the strength of the community division*" [Newman Girvan 2003]
- Optimization Problem: Find Qmax-- the maximum modularity of a given network.
- The decision the problem is NP-hard [Brandes *et. al.* 2007]

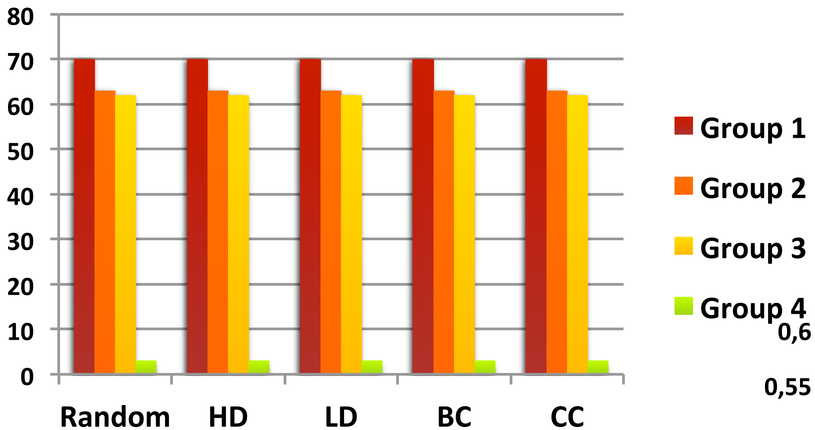
Vertex Perturbation

- Agglomerative methods –greedy heuristics for increasing modularity
- Value of modularity is affected by the order in which the vertices are processed
- The community membership also changes
- We use the CNM agglomerative method introduced by Clauset Newman and Moore (2004).

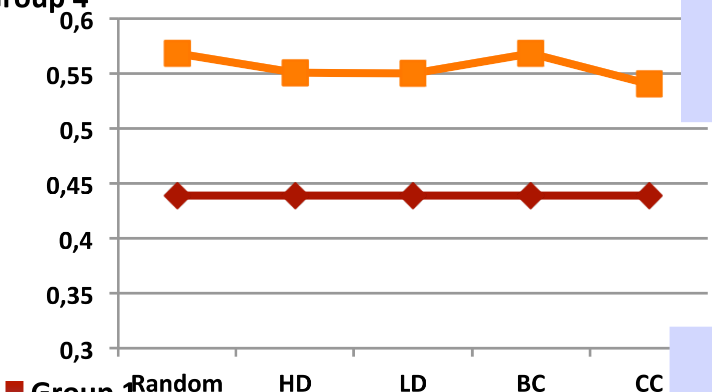
Agglomerative Clustering

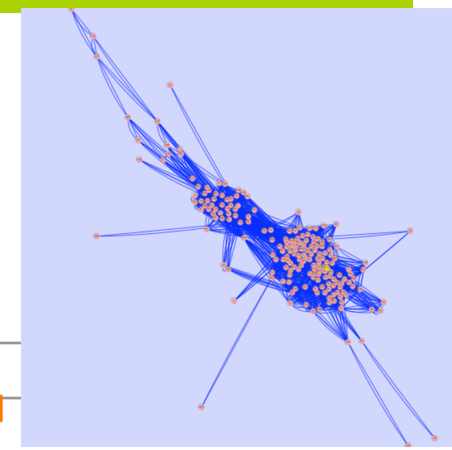


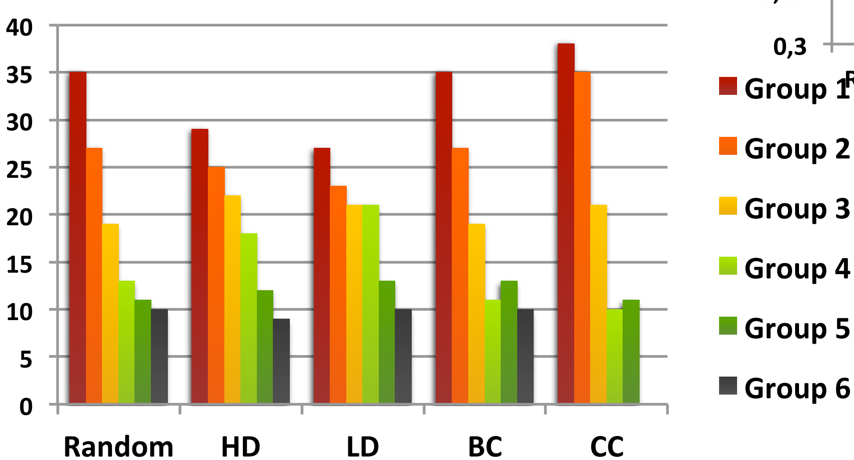
Effect of Vertex Perturbation



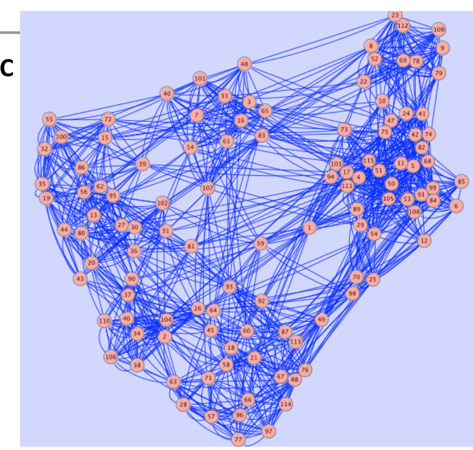
Jazz



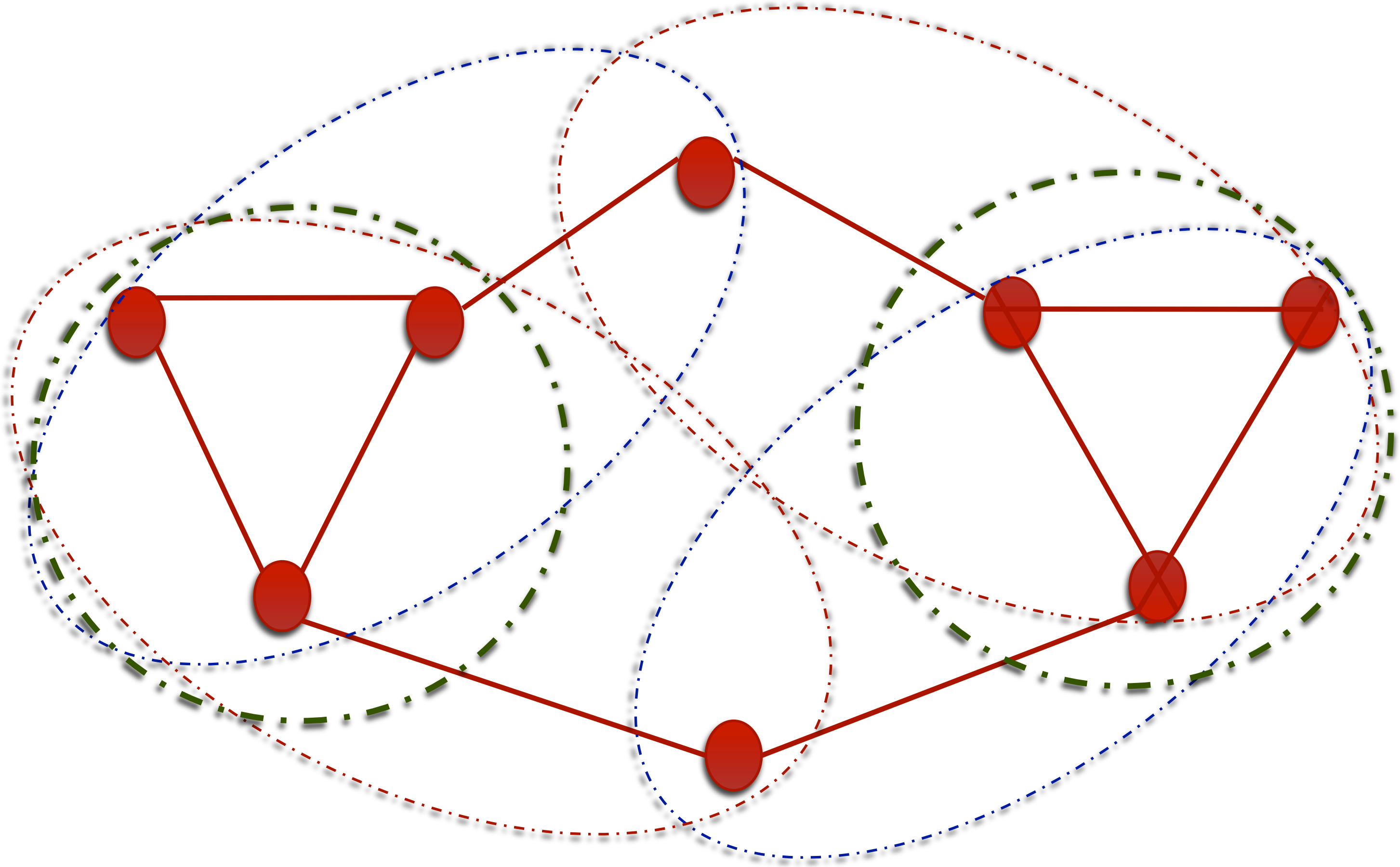




Football



Stable Communities



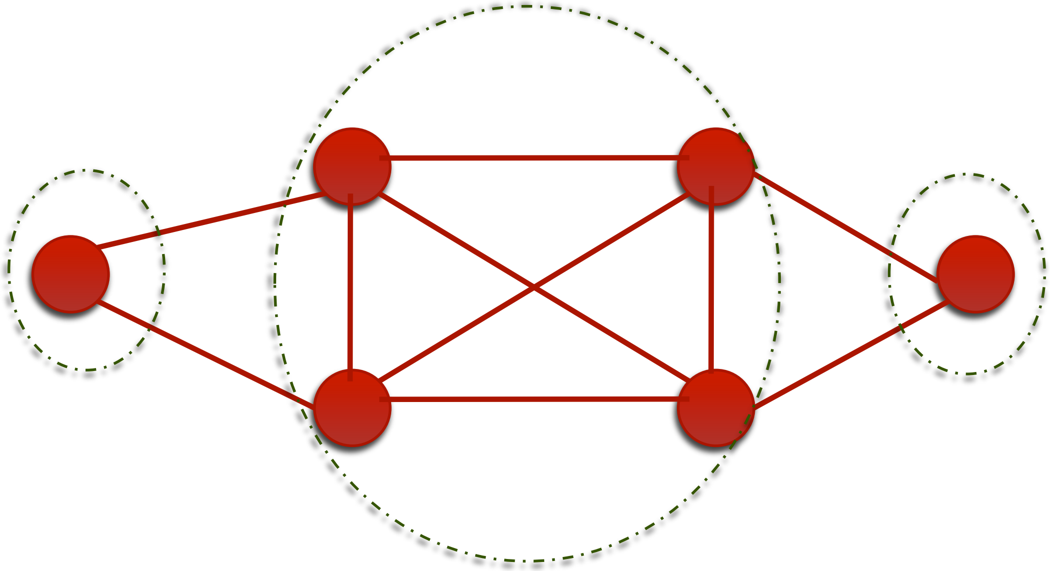
Maximum Modularity .25

Stable Communities

- Stable communities: groups of vertices that are always allocated to the same community independent of ordering or algorithm
- How do we identify groups of stable communities without executing first executing a community detection algorithm ?
- Define stable communities in terms of internal and external connections

First Approach

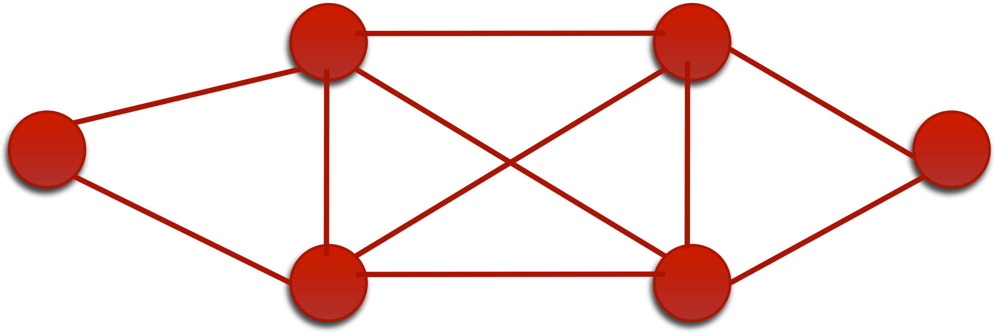
- Connections within the group >= Connections outside the group



Qmax=-.06

Second Approach

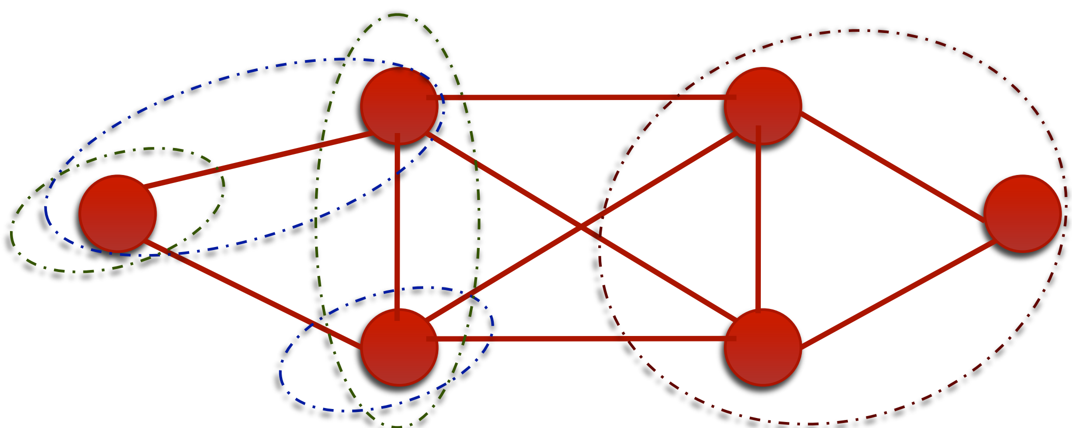
- Connections within each subset of vertices in the group >= Connections outside the group



Very difficult to find any groups that satisfy this condition

Third Approach

- Connections within each subset of vertices >= (Connections to any individual external group)/groups

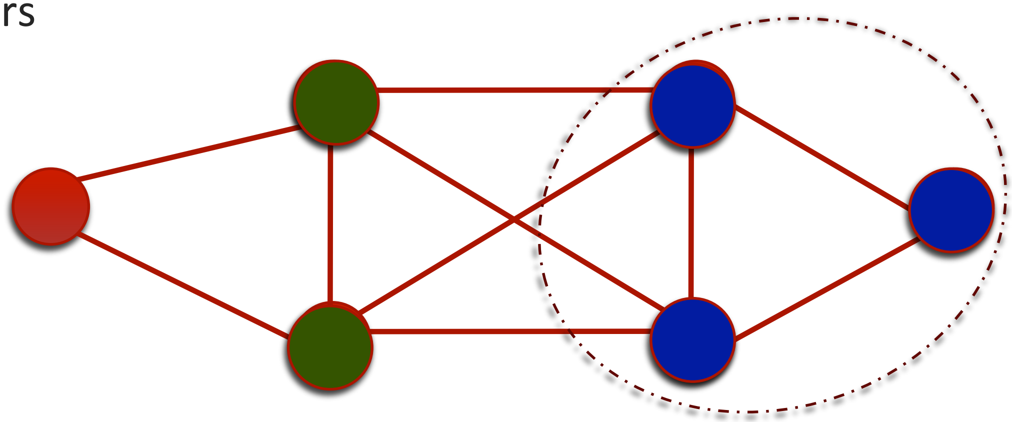


Qmax=-.02

Qmax=.02

Stable Communities of Strength K

- For each unallocated vertex vi
- Create subset S of vi and neighbors
- For all neighbors nj
  - xj=number of internal connections in S
  - yj= number of external connections in S where external neighbors are within distance k
- If xj>yj for all nj then S is a stable community
- Merge stable communities



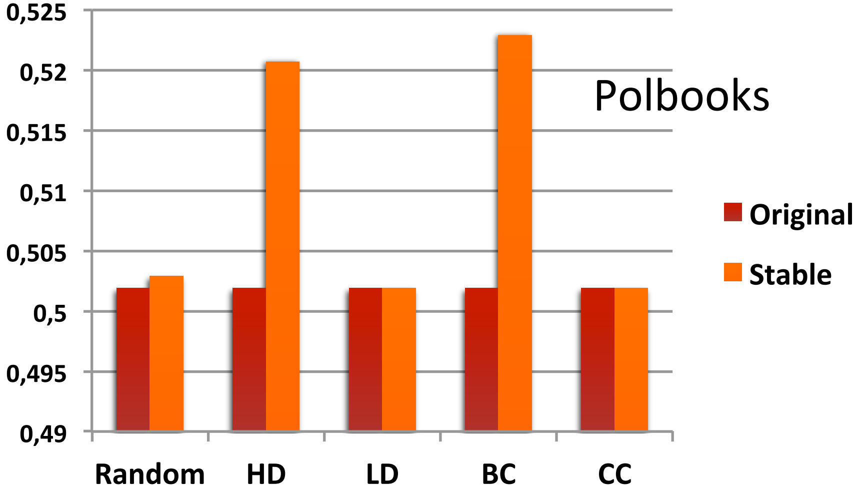
K=1; xj=3,2 yj=2

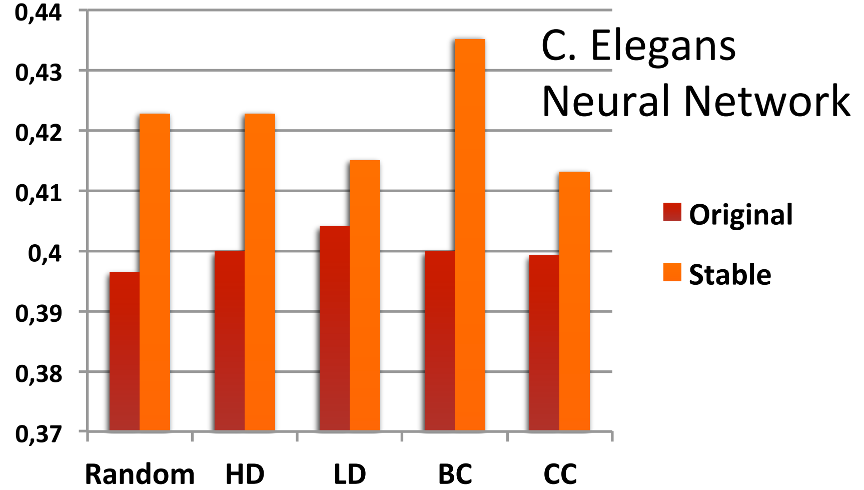
K=0; xj=3,2 yj=2 (1+1),2 (1+1)

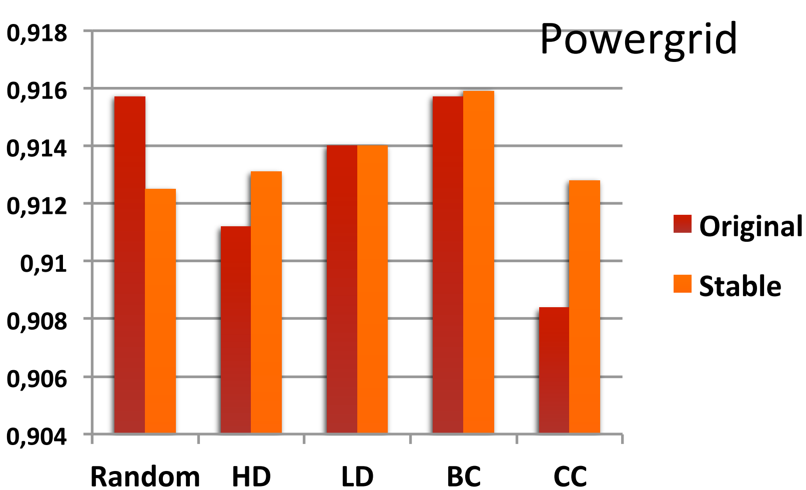
Preprocessing using Stable Communities

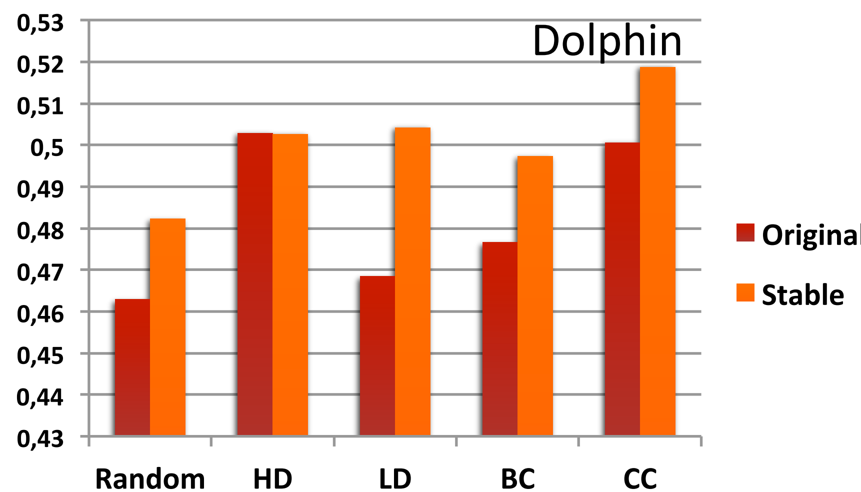
- We observed that the modularity value increases if stable communities are identified as a preprocessing step
  - Find stable communities
  - Combine them into communities
  - Execute agglomerative method for modularity maximization

Modularity Using Stable Communities

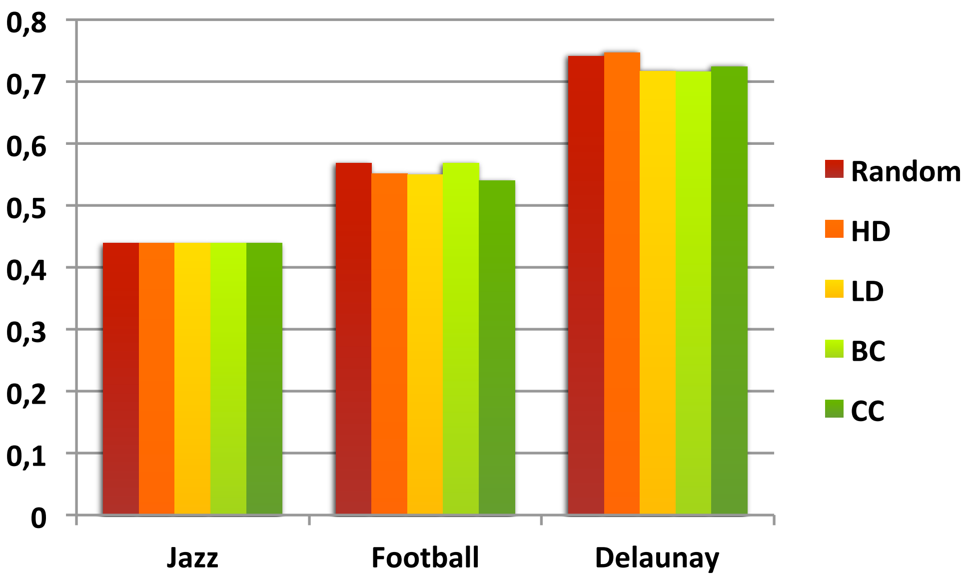






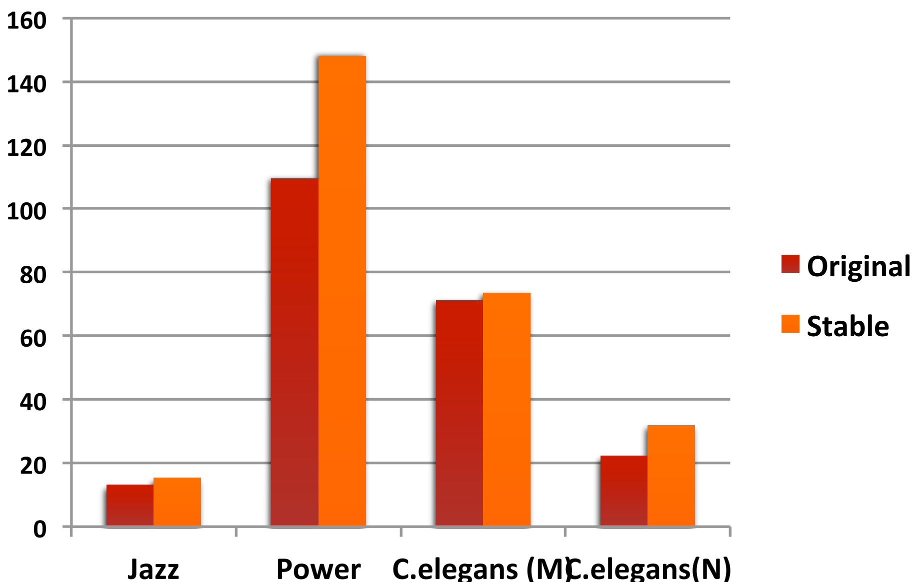


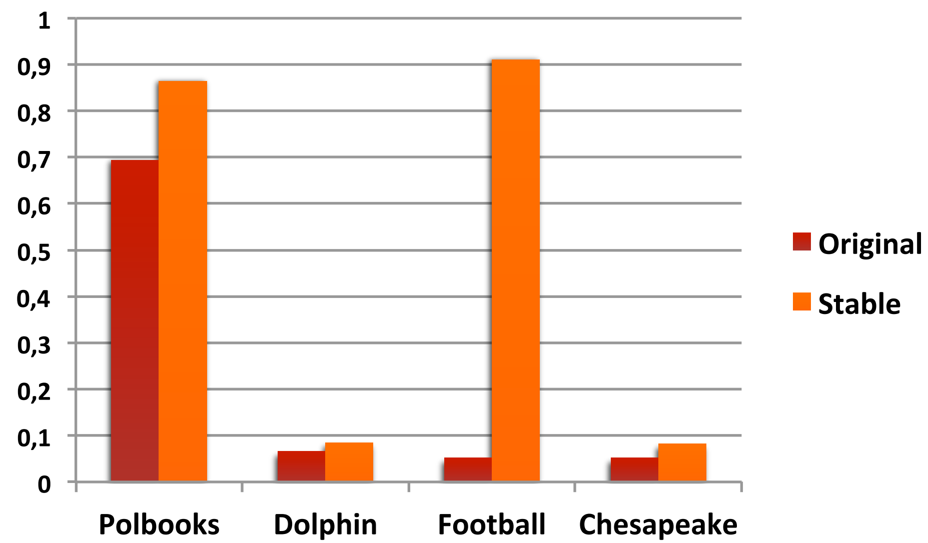
Modularity Using Stable Communities



- Jazz: 10 stable communities. Size 4-2
- Football: 1 stable community. Size 6
- Delaunay: No stable community

Execution Time





Execution time depends on the size and connectivity of the network

Also on the value of k----diameter of external communities

For bad examples execution time can get doubled without any benefit

Currently cannot handle > 5000 vertices

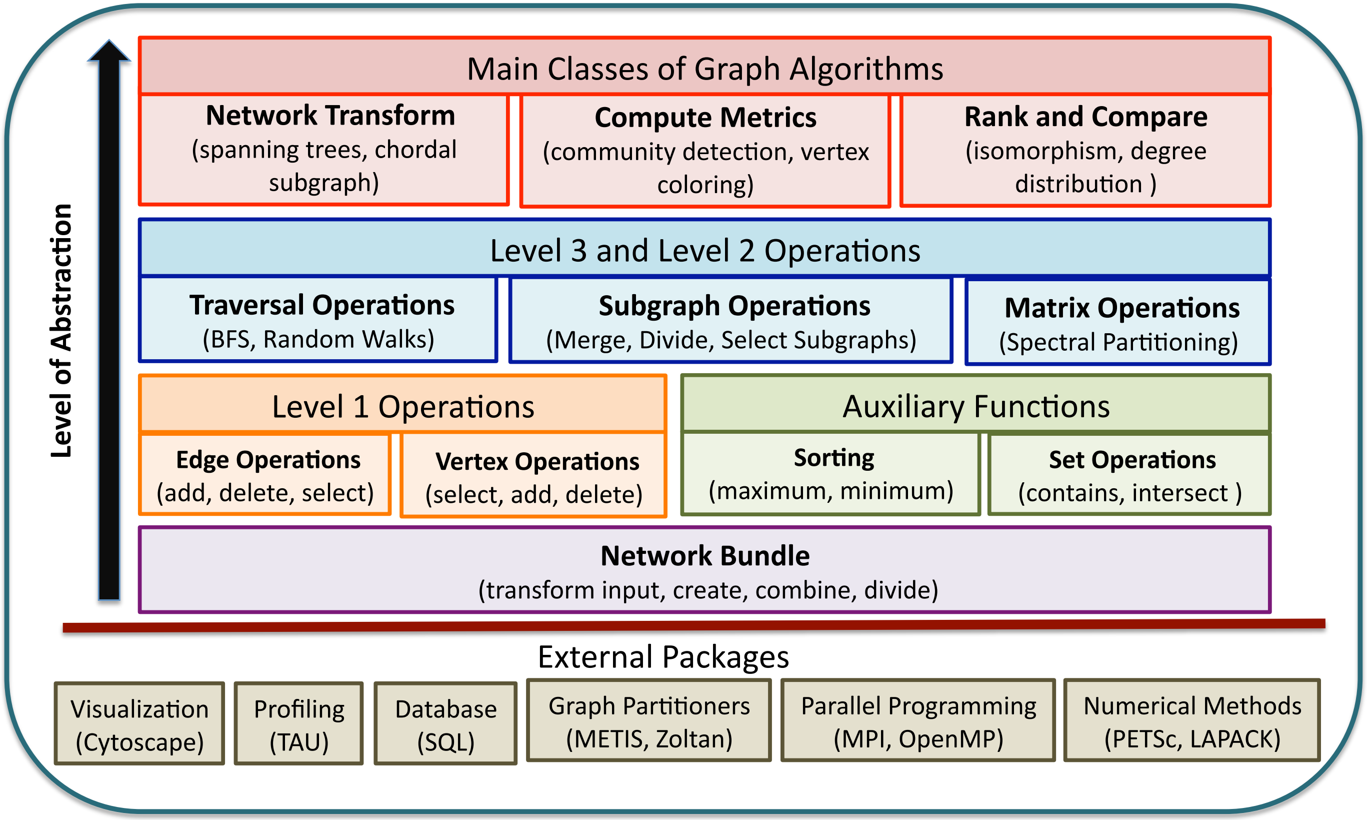
Practical Issues

- Quality of stable communities depends on
  - Value of k
  - Minimum size of community
- False positives can distort the results
- Currently is a preprocessing step for only agglomerative methods
- Final results depends on the underlying algorithm

Summary

- Measure the invariance of networks under ordering
- Improves the modularity if used as preprocessing
- Research Questions
  - How well does this address resolution limit, degeneracy of solutions ?
  - Can this be used to identify perturbation in networks ?
  - Faster and more accurate stable community detection

ESSENS



Extensible Scalable Software for Evolving NetworkS

Acknowledgements

- College of IST, University of Nebraska at Omaha
- FIRE, University of Nebraska at Omaha Sponsored Program Office