Applications of graph clustering

Sushmita Roy
sroy@biostat.wisc.edu
Computational Network Biology
Biostatistics & Medical Informatics 826
https://compnetbiocourse.discovery.wisc.edu

Nov 7th 2018
Unnormalized Graph Laplacian

- For a given graph \( G=\{V, E\} \)
- The unnormalized graph Laplacian is a \(|V| \times |V|\) matrix

\[ L = D - W \]
Unnormalized Graph Laplacian example

Example graph

Adjacency matrix \(W\)

\[
\begin{pmatrix}
1 & 2 & 3 & 4 \\
1 & 0 & 1 & 1 & 0 \\
2 & 1 & 0 & 1 & 0 \\
3 & 1 & 1 & 0 & 1 \\
4 & 0 & 0 & 1 & 0 \\
\end{pmatrix}
\]

Degree matrix \(D\)

\[
\begin{pmatrix}
1 & 2 & 3 & 4 \\
1 & 2 & 0 & 0 & 0 \\
2 & 0 & 2 & 0 & 0 \\
3 & 0 & 0 & 3 & 0 \\
4 & 0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

Laplacian \(L=D-W\)

\[
\begin{pmatrix}
1 & 2 & 3 & 4 \\
1 & 2 & -1 & -1 & 0 \\
2 & -1 & 2 & -1 & 0 \\
3 & -1 & -1 & 3 & -1 \\
4 & 0 & 0 & -1 & 1 \\
\end{pmatrix}
\]
Properties of the Laplacian

• For every vector $f$ in $\mathbb{R}^n$,
  $$f' L f = \frac{1}{2} \sum_{ij} w_{ij} (f_i - f_j)^2$$

• $L$ is symmetric and positive semi-definite
  $$f' L f \geq 0, \forall f \in \mathbb{R}^n$$

• The smallest eigen value of $L$ is 0 and its corresponding eigen vector is all 1s

• $L$ has $n$ non-negative eigen values
  $$0 = \lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$$
Number of connected components and the multiplicity of $\lambda=0$

- Let $G$ be an undirected graph with non-negative weights.
- Then the multiplicity, $k$, of the eigenvalue 0 of $L$ equals the number of connected components in the graph $A_1, \ldots, A_k$.
Number of connected components and L’s smallest eigen value

- To see why this is true, we use the property of an eigen vector, consider the case of one connected component
  - If $f$ is an eigen vector of $L$, then $Lf = \lambda f$
  - For eigen value 0, $Lf = 0$ (vector or all zeros)
- In addition we know

$$f' L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$$

- If $f$ is an eigen vector corresponding to eigen value =0, this must be 0
- The only way this can be 0 is if $f_i = f_j$ because $w_{ij}$ is non-zero
- This holds for all vertices connected by a path
- If all vertices are connected, then $f$ is a vector of constants
RECAP: Spectral clustering

• Based on the graph Laplacian

• Graph Laplacian \( L = D - W \)
  - \( D \) is the diagonal degree of matrix
  - \( W \) is the adjacency matrix

• Obtain the \( k \) eigen vectors associated with \( k \) smallest eigen values of \( L \)

• Represent each node as the \( k \)-dimensional vector

• Cluster nodes based on \( k \)-means clustering
Spectral clustering key steps

1. Adjacency matrix \((W)\)
2. Laplacian \((L)\)
3. First \(k\) eigen vectors

Result: \(k\) clusters
Application of graph clustering

• Finding higher-order Topologically Associated Domains from Hi-C data
• Disease module identification
• Similarity network fusion for aggregating data types on a genomic scale
Genome is organized into multiple organizational units

- Chromosomal territories through inter-TAD interactions
- Compartments and sub-compartments
- Topologically associated domains (TADs) and sub-TADs

A graph is a natural representation of a Hi-C dataset.
An overview of spectral clustering

Adjacency matrix

\[ A \]

Laplacian matrix

\[ L = D - A \]

First \( k \) eigenvectors

\( k \) clusters

D: Diagonal degree matrix
Does graph clustering help?

Silhouette index (higher is better)
Does graph clustering help?

A. Davies-Bouldin index (lower is better)

B. Silhouette index (higher is better)

C. Delta contact count (higher is better)

D. Number of enriched clusters (higher is better)

E. Heatmap showing correlations and clusters
Does graph clustering help?

Spectral (graph) clustering methods tend to do better on different measures.
Spectral clustering of Hi-C data of human ESC

(a) Spectral clusters for human ESC

(b) Distribution of chromosomes in clusters

Spearman correlation
Two main types of chromatin interaction modules
Topologically associated domains

The image shows a heat map of normalized interacting counts across different genomic regions on chromosome 6. The x-axis represents genomic positions in base pairs, and the y-axis shows the normalized interacting counts. The colors range from red (high counts) to white (low counts), with distinct peaks indicating regions of high interaction frequency. The graph also includes a bar chart indicating the presence of CTCF sites, with red bars representing CTCF association and blue bars representing non-association. The peaks in the graph correspond to areas enriched for CTCF binding, suggesting regions of high interaction and potential functional importance. The diagram is used to illustrate the distribution and patterns of interactions within these topologically associated domains.
Graph clustering to find TADs
Application of spectral clustering

• Finding higher-order Topologically Associated Domains from Hi-C data
• Disease module identification
• Similarity network fusion for aggregating data types on a genomic scale
DREAM community challenge for module identification

• A community challenge to assess algorithms for module identification across diverse molecular networks
• Six different networks
• Sub challenge 1: predict modules within a single network
• Sub challenge 2: predict modules across multiple networks.
• Evaluation: how many modules are associated with GWAS traits.

Choodbar et al., 2018 Bioarxiv
Overview of the DREAM disease module identification challenge

We launched an open-participation community challenge, where teams competed to predict groups of functionally related genes (i.e., modules) within diverse molecular networks. (A) The challenge comprised six networks, including protein-protein interaction, signaling, co-expression, cancer dependency, and homology-based gene networks. As the networks were all unpublished, we could anonymize them by removing the gene labels. This prevented participants from using existing knowledge of gene functions, thus enabling rigorous, blinded assessment.

(B) The aim of the challenge was to identify disease-relevant modules within the provided networks. Teams could participate in either or both sub-challenges: 42 teams predicted modules for individual networks (Sub-challenge 1) and 33 teams predicted integrated modules across multiple networks (Sub-challenge 2).

(C) The submitted modules were tested for association with complex traits and diseases using a comprehensive collection of 180 GWAS datasets. The final score for each method was the number of trait-associated modules that it discovered. Since GWAS are based on data completely different from those used to construct the networks, they can provide independent support for biologically relevant modules.

**Figure 1:**
- **A** Network compendium: 6 networks with varying degrees of connectivity.
- **B** Challenge: Anonymize networks to prevent using existing knowledge, enabling rigorous assessment.
- **C** GWAS compendium: 180 datasets covering various traits and diseases.

**Scoring:** Test modules for association with traits and diseases using gene scores.
Challenge organization

• Challenge was executed on Synapse
• Submissions accepted over a 2 month period where submitters could use benchmark data to assess and improve their predictions
• Final submissions were done on a separate GWAS dataset
Evaluation pipeline

• Six networks which were anonymized and given to challenge participants
• Consider modules of size 3-100 genes
• Assess modules based on GWAS association
Methods used

• 42 different methods from the following categories

A

- Kernel clustering
- Modularity optimization
- Random walk based
- Local methods
- Ensemble methods

Network

Genes

Transform network

Optimization algorithm

Random walk or diffusion process

Agglomerative process

Merge set of diverse predictions
Overview of results

The top teams used different approaches: the best performers ($K1$) developed a novel kernel approach leveraging a diffusion-based distance metric (Cao et al., 2013, 2014) and spectral clustering (Ng et al., 2001); the runner-up team ($M1$) extended different modularity optimization methods with a resistance parameter that controls the granularity of modules (Arenas et al., 2008); and the third-ranking team ($R1$) used a random-walk method based on multi-level Markov clustering with locally adaptive granularity to balance module sizes (Satuluri et al., 2010). Interestingly, teams employing the widely-used Weighted Gene Co-expression Network Analysis tool (WGCNA) (Langfelder and Horvath, 2008), which relies on hierarchical clustering to detect modules, did not perform competitively in this challenge (rank 35, 37 and 41).
Overview of results

Figure 2: Assessment of module identification methods

(A) Main types of module identification approaches used in the challenge: kernel clustering methods transform and cluster the network adjacency matrix; modularity optimization methods rely on search algorithms to find modular decompositions that maximize a structural quality metric; random walk-based methods take inspiration from diffusion processes over the network; local methods use agglomerative processes to grow modules from seed nodes; and ensemble methods merge alternative clusterings sampled either from stochastic runs of a given method or from a set of different methods. In addition, hybrid methods employ more than one of the above approaches and then pick the best modules according to a quality metric. See also Table 1.

(B) Final scores of the 42 module identification methods applied in Sub-challenge 1 for each of the six networks, as well as the overall score summarizing performance across networks (same method identifiers as in Table 1). Scores correspond to the number of unique trait-associated modules identified by a given method in a network (evaluated using the hold-out GWAS set at 5% FDR, see Methods). Ranks are indicated for the top ten methods. The last two rows show the performance of consensus predictions derived from the challenge submissions and randomly generated modules, respectively.

(C) Robustness of the overall ranking was evaluated by subsampling the GWAS set used for evaluation 1,000 times. For each method, the resulting distribution of ranks is shown as a boxplot. The rankings of method K1 are substantially better than those of the remaining teams (Bayes factor < 3, see Methods).

(D) Number of trait-associated modules per network. Boxplots show the number of trait-associated modules across methods, normalized by the size of the respective network. See also Fig. S1B.
Top performing method

• Use diffusion state distance (DSD) for each pair of vertices
• Convert into a similarity by passing it through the Gaussian kernel
• Apply spectral clustering
Other takeaways from disease module identification

- Co-expression and Protein-protein interaction network based modules were most informative.
- Top methods covered different categories.
  - But spectral clustering based methods worked best.
- Determining the right resolution can impact the results.
Application of spectral clustering

- Finding higher-order Topologically Associated Domains from Hi-C data
- Disease module identification
- Similarity network fusion for aggregating data types on a genomic scale
Similarity network fusion for aggregating data types on a genomic scale

• This paper had two goals:
  – Integrate different types of data using a network-based approach
  – Identify groups of samples representing integrated data types
• Recent high throughput technologies have made it possible to collect many different types of genomic data for individual patients
• How do we combine patient data to describe a disease?
• This is challenging because of the following issues:
  – Noisy samples
  – Small number of samples than variables
  – Complimentary nature of the data
Similarity Network Fusion

• Given N different types of measurements for different individuals

• Do
  – Construct a similarity matrix of individuals for each data type
  – Integrate the networks using a single similarity matrix using an iterative algorithm
  – Cluster the network into groups of individuals
Similarity network fusion (Nodes are patients, edges represent similarities).
Defining a similarity graph over patient samples

- For each data type, create a weighted graph, with vertices corresponding to patients.
- Let $x_i$ and $x_j$ denote the measurements of patients $i$ and $j$.
- Edge weights, $W(i, j)$ correspond to how similar patient $i$ is to patient $j$ based on $x_i$ and $x_j$.

$$W(i, j) = \exp\left( - \frac{\rho^2(x_i, x_j)}{\mu \epsilon_{i,j}} \right)$$

- Euclidean distance
- Hyper-parameter
- Scaling term (average of the distance between each node and its neighborhood)
Creating a fused matrix

- Define two matrices for each data type
- A full matrix: normalized weight matrix
  \[ P(i, j) = \begin{cases} \frac{W(i, j)}{2\sum_{k \neq i} W(i, k)}, & j \neq i \\ 1/2, & j = i \end{cases} \]
- A sparse matrix (based on k nearest neighbors or each node)
  \[ S(i, j) = \begin{cases} \frac{W(i, j)}{\sum_{k \in N_i} W(i, k)}, & j \in N_i \\ 0, & \text{otherwise} \end{cases} \]

This makes the assumption that the local similarities are the most reliable
Iterate for fusion

- Input $m$ data types
- Construct $W^{(v)}$ for each data type $v$
- Construct dense matrix $P^{(v)}$ and sparse matrix $S^{(v)}$
- At each iteration, update the dense similarity matrix of one data type using the similarity matrix of the other data type
Iteration with m=2 data types

For iteration $t+1$

Update similarity matrix of data type 1

\[ P_{t+1}^{(1)} = S^{(1)} \times P_t^{(2)} \times (S^{(1)})^T \]

Update similarity matrix of data type 2

\[ P_{t+1}^{(2)} = S^{(2)} \times P_t^{(1)} \times (S^{(2)})^T \]

Update similarity matrix of data type 1 using weight matrix from data type 2 and vice-versa
What is going on in the iteration step

\[ P_{t+1}^{(1)}(i, j) = \sum_{k \in N_i} \sum_{l \in N_j} S^{(1)}(i, k) \times S^{(1)}(j, l) \times P_t^{(2)}(k, l) \]

We are updating the similarity matrix using the most confident common neighbors of \( i \) and \( j \)
Extending to m>2 data types

\[ P^{(\nu)} = S^{(\nu)} \times \left( \frac{\sum_{k \neq \nu} P^{(k)}}{m - 1} \right) \times (S^{(\nu)})^T, \nu = 1, 2, \ldots, m \]

Just average over all other data types
SNF termination

• After repeating the iterative updates for \( t \) steps, final similarity matrix is

\[
P = \frac{1}{m} \sum_{k=1}^{m} P_t^k
\]

• This is then clustered using spectral clustering
Application of SNF to Glioblastoma

• Contradicting information about subtypes depending upon the type of data used
• Glioblastoma dataset
• Three data types among 215 patients
  • DNA methylation (1491 genes)
  • mRNA (12,042 genes)
  • miRNA (534 miRNAs)
SNF application to GBM identifies 3 subtypes

DNA methylation

mRNA expression

miRNA expression
Validation of SNF identified subtypes

Subtypes are associated with patient populations of different survival. Blue curve (subtype 3) are patients with more favorable prognosis.
Key points of graph clustering algorithms

• Flat or hierarchical clustering

• Algorithms differ in
  – how they define the similarity/distance measure
    • Local topology measures
    • Global measures
  – Whether the algorithm takes as input the number of clusters or the goodness of clusters (e.g. the approximate cluster algorithm)
References


