Robust Multi-Network Clustering via Joint Cross-Domain Cluster Alignment

Rui Liu† *, Wei Cheng‡ †, Hanghang Tong‡, Wei Wang§ and Xiang Zhang*

*Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106
Email: {rui.liu4, xiang.zhang}@case.edu
†Department of Computer Science, University of North Carolina at Chapel Hill, NC 27599
Email: weicheng@cs.unc.edu
‡School of Computing, Informatics, Decision Systems Engineering, Arizona State University, Tempe, AZ 85281
Email: htong6@asu.edu
§Department of Computer Science, University of California, Los Angeles, CA 90095
Email: weiwang@cs.ucla.edu

Abstract—Network clustering is an important problem that has recently drawn a lot of attentions. Most existing work focuses on clustering nodes within a single network. In many applications, however, there exist multiple related networks, in which each network may be constructed from a different domain and instances in one domain may be related to instances in other domains. In this paper, we propose a robust algorithm, MCA, for multi-network clustering that takes into account cross-domain relationships between instances. MCA has several advantages over the existing single network clustering methods. First, it is able to detect associations between clusters from different domains, which, however, is not addressed by any existing methods. Second, it achieves more consistent clustering results on multiple networks by leveraging the duality between clustering individual networks and inferring cross-network cluster alignment. Finally, it provides a multi-network clustering solution that is more robust to noise and errors. We perform extensive experiments on a variety of real and synthetic networks to demonstrate the effectiveness and efficiency of MCA.

I. INTRODUCTION

Networks (or graphs) are widely used in representing relationships between instances, in which each node corresponds to an instance and each edge depicts the relationship between a pair of instances. Network clustering (or graph clustering) [1]–[3] has become an effective means in discovering modules formed by closely related instances in such networks, which may in turn reveal functional structure of the networks. Recently, the attention has moved from clustering in a single homogeneous network (built on instances from one domain) to joint clustering on multiple heterogeneous networks (from different but related domains), due to obvious reasons: integrating information from different but related domains not only may help to resolve ambiguity and inconsistency in clustering outcome, but also may discover and leverage strong associations between clusters from different domains. Consequently, these multi-view network clustering methods [3], [4] are able to substantially improve the clustering accuracy. For example, millions of genetic variants on human genome have been reported to be disease related, most of which are in the form of single nucleotide polymorphism (SNP). These SNPs do not function independently. Instead, a set of SNPs may play joint roles in a disease. Such interactions between SNPs can be modeled by a SNP interaction network.

† These authors contributed equally to this work

Fig. 1. An exemplar SNP interaction network and gene interaction network in an eQTL study

Fig. 1 shows an exemplar SNP interaction network $G_1$ of 17 SNPs on the left, in which nodes are SNPs and weighted edges represent interactions between SNPs. Even though the underlying biological processes are complex and only partially solved, it is well established that SNPs may alter the expression levels of related genes which may in turn have a cascading effect to other genes, e.g., in the same biological pathways [5]. The interactions between genes can be measured by correlations of gene expressions and represented by a gene interaction network. Fig. 1 shows an exemplar gene interaction network of 20 genes on the right, in which nodes are genes and weighted edges represent interactions between genes. These two networks are heavily related because of the (complicated) relationships between SNPs and genes, as demonstrated in many expression quantitative trait loci (eQTL) studies. These cross-domain relationships are represented by dotted edges between SNPs and genes in Fig. 1. The strength of such relationship is coded by the edge weight. It is evident that a joint analysis becomes essential in these related domains.

Despite the success of previous approaches in network clustering, they still suffer from two common limitations. First, existing methods usually assume that information collected in different domains are for the same set of instances. Thus, the cross-domain instance relationships are strictly one-to-one correspondence. This assumption may not hold in many applications. More often than not, data instances (e.g., SNPs) in one domain may be related to multiple instances (e.g., genes) in another domain. Methods that can account for many-to-many cross-domain relationships are in need [6]. Second, existing approaches tend to focus on network clustering and
ignore any associations that may be exhibited between clusters from different domains. However, “alignment” between clusters from multiple domains may provide a more comprehensive depiction of the whole system. For example, a cluster of SNPs may jointly regulate the expressions of a cluster of genes, which may be revealed by cluster level associations. Fig. 1 shows 2 SNP clusters: A (including SNPs \{1, 2, 3, 4\}) and B (including SNPs \{12, 13, 14, 16\}), and 3 gene clusters: C (including genes \{a, b, c, d\}), D (including genes \{p, q, r, s\}) and E (including genes \{i, j, k, m\}). As summarized in Table I, SNP cluster A is strongly associated with gene cluster C and SNP cluster B is strongly associated with gene cluster D. Gene cluster E is not strongly associated with any SNP cluster. Although we are given cross-domain associations at the instance level, it is still nontrivial to discover cross-domain associations at the cluster level, especially in the presence of noise. Our goal is to discover such strong associations between pairs of clusters from different domains simultaneously when we perform network clustering.

| Cross-domain associations between SNP clusters and gene clusters in Fig. 1. |
|----------------------|----------------------|
| Association          | 1st cluster pair     | 2nd cluster pair |
| SNP interaction network C | \{12, 13, 14, 16\} | \{1, 2, 3, 4\} |
| gene interaction network C | \{p, q, r, s\} | \{a, b, c, d\} |

In this paper, we propose a robust approach, MCA (Multi-network Clustering via cluster Alignment), to detect network clusters in multiple domains and their cross-domain associations. In addition to the advantages discussed above, the duality between clustering in individual networks and inferring cross-network cluster alignment enables mutual reinforcement when both tasks are performed simultaneously. As a result, MCA can effectively filter noise and resolve ambiguities in individual networks, and achieve much higher accuracy in detecting network clusters and their cross-domain associations. It also employs a sparsity regularizer on the cluster alignment to provide additional robustness to noise in the prior cross-domain (instance-level) relationships.

Our contributions are summarized as follows.

- To the best of our knowledge, little prior work has studied the problem of cross-domain cluster association detection. In this paper, we propose and investigate this novel problem under the multi-domain setting. The problem is essential to a wide range of applications.

- We develop a framework, MCA, based on nonnegative matrix tri-factorization to simultaneously cluster instances within each domain and reveal the associations between clusters from different domains. Clustering and cluster association discovery could mutually enhance each other. We provide rigorous theoretical analysis of MCA in terms of its correctness, convergence and complexity.

- We evaluate MCA by extensive experiments on both synthetic and real datasets. The experimental results demonstrate that MCA is superior to existing approaches in both clustering accuracy and cluster association accuracy.

II. Related Work

To the best of our knowledge, this is the first work to “align” clusters across multiple domains. Existing work on network clustering primarily focused on clustering in a single network [7], [9]. In [9], the authors pioneered a graph partitioning algorithm using normalized cut. Spectral clustering has gained popularity as an efficient clustering algorithms in recent years [11]. It is simple to implement since its computational burden is primarily on the computation of eigenvectors which has been studied in-depth in numerical analysis. In [7], the authors proposed a framework to detect communities in a network based on modularity. Some other multi-domain graph (network) clustering methods [3] focus on improving the clustering accuracy within each domain utilizing information from other domains. The cross-domain instance relationships are only used to enhance the clustering result within each domain. They do not capture associations between clusters across multiple domains. Multi-domain data are inherently heterogeneous. Networks constructed from multiple related domains can be transformed into a heterogeneous information network, on which clustering may be performed [12]. However, this pioneer work focused on ranking-based clustering which ranks clusters on a pre-specified target type (domain). This is different from our goal of performing clustering within and across domains. In addition, some methods on co-clustering also make use of Nonnegative Matrix Tri-Factorization (NMTF) and graph regularizer. Co-clustering [13], DNMTF [14] and RCC [15] were originally designed to improve the clustering accuracy on documents by clustering rows and columns of a term-document matrix simultaneously [16]. They usually pay special attentions to the duality of clustering of terms and clustering of documents. The cross-domain cluster associations are not explicitly considered by co-clustering methods, even though some of these methods may be adapted to derive information about cross-domain cluster associations. They are either incapable of handling multi-network setting or sensitive to noise, since they were not designed for network clustering.

III. Multi-Network Clustering via Cross-Domain Cluster Alignment

In this section, we discuss the problem definition and our proposed algorithm MCA.

A. Problem Definition

Suppose that we have \(N\) domains \(\{D_1, D_2, \ldots, D_N\}\). Instances and their relationships within each domain are represented by a network \(G_p\), \(1 \leq p \leq N\). Let \(A_p\) be the affinity/adjacency matrix of \(G_p\). It is possible that some instances in domain \(D_p\), \(1 \leq p \leq N\) may be related to some instances in domain \(D_q\), \(1 \leq q \leq N\). These cross-domain relationships between instances can be represented by a matrix \(W_{pq}\). Important notations are listed in Table II. More often than not, the matrix \(W_{pq}\) is derived from prior knowledge and may be incomplete and noisy. Our goal is to integrate these cross-network instance relationships into the task of multi-network clustering and infer cross-network associations between clusters. We formulate this problem as an optimization problem that generates clustering and cluster associations simultaneously. We now discuss them in detail.
TABLE II. SUMMARY OF SYMBOLS AND THEIR MEANINGS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>The number of domains</td>
</tr>
<tr>
<td>D_p</td>
<td>The p-th domain</td>
</tr>
<tr>
<td>n_p</td>
<td>The number of instances in D_p</td>
</tr>
<tr>
<td>k_p</td>
<td>The number of clusters in D_p</td>
</tr>
<tr>
<td>Y_p</td>
<td>The network representing relationship among instances in D_p</td>
</tr>
<tr>
<td>A_p</td>
<td>The affinity/adjacency matrix of Y_p</td>
</tr>
<tr>
<td>W_pq</td>
<td>The cross-domain relationship between instances from D_p and D_q</td>
</tr>
<tr>
<td>H_p</td>
<td>The cluster assignment matrix in D_p</td>
</tr>
<tr>
<td>S_pq</td>
<td>The cross-domain alignment matrix between D_p and D_q</td>
</tr>
</tbody>
</table>

For simplicity, we begin with 2 domains. For clustering in domain D_1, we want to minimize the following objective function

\[
\sum_{ij} A_{1,ij} \|H_{1,ij} - H_{1,ij}\|^2_2, \quad \text{s.t. } H_{1}^T H_{1} = I, H_{1} \geq 0
\]

where H_1 is the cluster assignment matrix in domain D_1 and H_{1,ij} means the i-th row of matrix H_1. H_{1,ij,k} can be viewed as the probability that the i-th instance in domain D_1 belongs to the k-th cluster of this domain. A similar objective function can be applied to clustering in domain D_2.

In order to capture the cross-network cluster associations, we adopt the co-clustering strategy that minimizes the following objective function

\[
\|W_{12} - H_1 S_{12} H_2^T\|^2_F + \eta_1 \|S_{12}\|_1, \quad \text{s.t. } H_{1}^T H_{1} = I, H_{2}^T H_{2} = I, H_{1} \geq 0, H_{2} \geq 0
\]

This objective function is the Sparse Nonnegative Matrix Tuckerization. With orthogonality constraints on H_1 and H_2, it is equivalent to running K-means co-clustering on W_{12} [13]. S_{12} is the cross-domain alignment matrix, depicting the alignment of the clusters from the two domains. Because W_{12} may contain noise, we employ the \ell_1-norm on S_{12} to suppress the influence of any inconsistencies in W_{12}.

Combining together the above two parts, we have the following optimization problem

\[
\begin{align*}
\min_{H_1,H_2,S_{pq}} & \|W_{12} - H_1 S_{12} H_2^T\|^2_F + \alpha_1 \sum_{ij} A_{1,ij} \|H_{1,ij} - H_{1,ij}\|^2_2 \\
& + \alpha_2 \sum_{ij} A_{2,ij} \|H_{2,ij} - H_{2,ij}\|^2_2 + \eta_1 \|S_{p}\|_1 \\
\text{s.t. } & H_{1}^T H_{1} = I, H_{2}^T H_{2} = I, H_{1} \geq 0, H_{2} \geq 0
\end{align*}
\]

(1)

where \alpha_1, \alpha_2 and \eta_1 are parameters that balance different between terms, whose values can be determined via cross validation.

By simplifying it, we obtain

\[
\begin{align*}
\min_{H_1,H_2,S_{pq}} & \|W_{12} - H_1 S_{12} H_2^T\|^2_F + \alpha_1 \text{Tr}(H_1^T \Theta_1 H_1) \\
& + \alpha_2 \text{Tr}(H_2^T \Theta_2 H_2) + \eta_1 \|S_{p}\|_1 \\
\text{s.t. } & H_{1}^T H_{1} = I, H_{2}^T H_{2} = I
\end{align*}
\]

(2)

where \Theta_1 and \Theta_2 are the Laplacian Matrices of G_1 and G_2, respectively.

Now we come back to the multi-domain case. Let W_pq be the matrix defining instance level relationships between domain D_p and domain D_q. Here we assume W_pq = W_q^T. Similarly, we use S_pq to represent the cross-domain cluster alignment matrix between domain D_p and domain D_q and we have S_pq = S_{pq}^T. The optimization problem in Eq.(2) can be naturally extended to the following multi-domain case.

\[
\begin{align*}
\min_{H_p,S_{pq}} & \|W_{pq} - H_p S_{pq} H_q^T\|^2_F + \\
& \sum_{p \neq q} \alpha_p \text{Tr}(H_p^T \Theta_p H_p) + \\
& \sum_{p \neq q} \eta_{pq} \|S_{pq}\|_1, \quad \text{s.t. } H_{p}^T H_{p} = I, \forall p
\end{align*}
\]

(3)

B. Learning Algorithm

In this section, we present the learning algorithm, MCA, to solve the optimization problem in Eq. (3). Since the objective function is not jointly convex with respect to all variables, we adopt an alternating optimization scheme. Specifically, each time we optimize the objective with respective to one variable while fixing others. The following two theorems set the foundation for our algorithm. Their correctness and convergence are guaranteed, which will be proven later.

**Theorem 1**: While fixing other variables, the objective function in Eq. (3) will monotonically decrease every time we update S_{pq} according to Eq. (4) until convergence.

\[
S_{pq} \leftarrow S_{pq} \circ \sqrt{H_p^T W_{pq} H_q \left( H_p^T H_p S_{pq} H_q^T H_q + \frac{1}{\eta_{pq}} \right)^{-1}}
\]

(4)

**Theorem 2**: While fixing other variables, the objective function in Eq. (3) will monotonically decrease every time we update H_p according to Eq. (5) until convergence.

\[
H_p \leftarrow H_p \circ \sqrt{\sum_{q \neq p} (W_{pq} H_q S_{pq}^T) + \alpha_p (\Theta_p H_p)} \\
H_p H_p^T \left( \sum_{q \neq p} (W_{pq} H_q S_{pq}^T) + \alpha_p (\Theta_p H_p) \right)^{-1}
\]

(5)

where \Theta_p is the negative part of \Theta_p, i.e., \Theta_p^- = (|\Theta_p| - \Theta_p)/2.

Note that \circ, (\cdot) and \sqrt{(\cdot)} are element-wise multiplication, division and square root, respectively. Based on Theorems 1 and 2, we develop an iterative updating algorithm summarized in Algorithm 1.

C. Correctness Analysis

In this section, we give the correctness analysis of the updating rules in Theorem 1, according to the Karush-Kuhn-Tucker (KKT) condition. The proof of updating rules for Theorem 2 is similar and hence omitted here.

Define the Lagrangian function with respect to S_{pq} as

\[
L(S_{pq}) = -2 \text{Tr}(H_p^T W_{pq} H_q S_{pq}^T) + \text{Tr}(H_p S_{pq} H_q^T H_q S_{pq}^T H_p^T) + \eta_{pq} S_{pq} - \text{Tr}(A_{pq} S_{pq}^T)
\]

(6)

where A_{pq} is a symmetric matrix whose entries are the lagrangian multipliers. Note that \|S_{pq}\|_1 becomes S_{pq} in Eq. (6) because S_{pq} \geq 0. We also omit all constant terms with
Algorithm 1: The MCA Algorithm

```latex
\textbf{input} : W_{pq}, A_p (1 \leq p, q \leq N, p \neq q) \\
\textbf{output} : H_p, S_{pq} (1 \leq p, q \leq N, p \neq q) \\
\textbf{begin} \\
\text{Random initialization of } H_p \text{ and } S_{pq} \\
(1 \leq p, q \leq N, p \neq q); \\
\Theta_p = D_A_p - A_p, (1 \leq p \leq N) \text{ where } D_A_p \text{ is the degree matrix of } A_p; \\
\text{while no convergence} do \\
\text{for } p = 1 \text{ to } N \text{ do} \\
\text{for } q = 1 \text{ to } N \text{ do} \\
\text{if } p \neq q \text{ then} \\
\text{update } S_{pq} \leftarrow S_{pq} \circ \sqrt{\frac{(H_p^T W_{pq} H_q)}{(H_p^T H_p S_{pq} H_q^T H_q^T) + \frac{1}{2} \eta_{pq}}}; \\
\text{for } p = 1 \text{ to } N \text{ do} \\
\text{update } H_p \leftarrow H_p \circ \sqrt{\frac{\sum_{q \neq p} (W_{pq} H_q S_{pq} + \alpha_p (\Theta_p H_p))}{(H_p^T H_p S_{pq} H_q^T H_q^T) + \alpha_p (\Theta_p H_p))}}; \\
\text{end while} \\
\textbf{end} \\
```

D. Convergence Analysis

In this subsection, we prove the guarantee of convergence using an auxiliary function [17].

**Definition 1**: \([17] \ Z(h, h') \) is an auxiliary function for \(f(h)\) if the conditions

\[ Z(h, h') \geq f(h), Z(h, h) = f(h) \] (11)

are satisfied for any given \(h, h'\).

**Lemma 1**: If \(Z\) is an auxiliary function for \(f\), then \(f\) is non-increasing under the update [17]

\[ h^{(t+1)} = \text{arg min}_h Z(h, h^{(t)}) \] (12)

**Proof**: \(f(h^{(t+1)}) \leq Z(h^{(t+1)}, h^{(t)}) \leq Z(h^{(t)}, h^{(t)}) = f(h^{(t)})\).

The auxiliary function with respect to \(S\) is

\[ Z(S_{pq}, S'_{pq}) = -2 \sum_{i,j} (H_p^T W_{pq} H_q)_{ij} S'_{pq,ij} (1 + \log \frac{S_{pq,ij}}{S'_{pq,ij}}) + \sum_{i,j} (\frac{H_p^T H_p S_{pq} H_q H_q^T}{S_{pq,ij}} - 2 \frac{S_{pq,ij}^2}{S'_{pq,ij}}) \] (13)

\[ \frac{\partial Z}{\partial S_{pq,ij}} = -2 (H_p^T W_{pq} H_q)_{ij} S'_{pq,ij} + \sum_{i,j} (\frac{H_p^T H_p S_{pq} H_q H_q^T}{S_{pq,ij}} - 2 \frac{S_{pq,ij}^2}{S'_{pq,ij}}) S_{pq,ij} + \eta_{pq} \sum_{i,j} \frac{S'_{pq,ij}^2}{S_{pq,ij}} \] (14)

\[ S_{pq,ij} = S'_{pq,ij} \sqrt{(H_p^T W_{pq} H_q)_{ij} + \frac{1}{2} \eta_{pq}} \] (15)

Similarly, the auxiliary function with respect to \(H_p\) is

\[ Z(H_p, H'_p) = -2 \sum_{q \neq p} (W_{pq} H_q S'_{pq})_{ij} H'_{pi,j} (1 + \log \frac{H_p}{H'_p}) + \sum_{q \neq p} (\frac{H'_p S'_{pq} H'_q S'_{pq}^T}{H'_p^T H'_p} - 2 \frac{H'_p S'_{pq}^2}{H'_p}) \] (16)
number of clusters in a domain, which leads to
\[ \sum \text{instances} \] is much larger than the overall time complexity is
\[ O(\text{Iter}(\tilde{n}^2k + \tilde{n}\tilde{k}^2 + k^3 + N\tilde{n}\tilde{k}^2)). \]

E. Complexity Analysis

With proper order of multiplication, updating \( S \) and \( H \) once require \( O(\tilde{n}^2k + \tilde{n}\tilde{k}^2 + k^3) \) and \( O(N\tilde{n}\tilde{k}^2 + \tilde{n}^2k) \), where \( \tilde{n} = \max \{ n_i \} \) is the largest number of instances among all domains and \( \tilde{k} = \max \{ k_i \} \) is the largest number of clusters among all domains. If the number of iterations is \( \text{Iter} \), the overall time complexity is \( O(\text{Iter}(\tilde{n}^2k + \tilde{n}\tilde{k}^2 + k^3 + N\tilde{n}\tilde{k}^2)). \) In practice, the number of instances is much larger than the number of clusters in a domain, which leads to \( \tilde{n} \gg \tilde{k} \). In this scenario, the overall time complexity of MCA can be simplified to \( O(\text{Iter}(\tilde{n}^2k + N\tilde{n}\tilde{k}^2)). \)

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of MCA on both synthetic and real datasets. To the best of our knowledge, there is no previous method that was specifically designed to discover cluster associations. Some co-clustering methods might be adapted to infer cluster associations. We compare our method MCA with three well-known co-clustering methods: Non-negative Matrix Tri-Factorization proposed by Chris Ding (denoted as NMTF_Chris in our paper) [13], Graph Dual Regularization Non-negative Matrix Tri-Factorization (DNMTF) [14] and Robust Co-Clustering (RCC) [15]. The parameters of each algorithm are tuned using a 5-fold cross validation. Other co-clustering methods are not compared because they are either not suitable for network clustering or do not impose nonnegativity on association matrix \( S \). The nonnegativity constraint on \( S \) is essential to ensure the result interpretability in this problem setting.

A. Evaluation Metrics

We evaluate our results in two perspectives: clustering accuracy within each domain and cluster association accuracy across domains.

1) Clustering Accuracy: We use the widely used normalized mutual information (MI) metric to evaluate the clustering accuracy in each domain. For any domain \( D \), assume that \( C = \{ c_i, i = 1, 2, \ldots, k \} \) is the clustering result where \( c_i \) is the \( i \)-th cluster. Let \( T = \{ t_i, i = 1, 2, \ldots, k \} \) be the ground truth where \( t_i \) is the \( i \)-th cluster. The normalized MI is defined as
\[ MI(C, T) = \frac{MI(C, T)}{\max(H(C), H(T))} \]

where \( H(C) \) and \( H(T) \) are the entropies for clustering \( C \) and \( T \) and \( MI(C, T) \) is the mutual information between \( C \) and \( T \).
\[ MI(C, T) = \sum_{c_i \in C, t_j \in T} p(c_i, t_j) \ln \frac{p(c_i, t_j)}{p(c_i)p(t_j)} \]

where \( p(c_i) \) is the percentage of instances contained in \( c_i \) and \( p(c_i, t_j) \) is the percentage of instances contained in the intersection of \( c_i \) and \( t_j \).

2) Cluster Association Accuracy: To evaluate the cross-network cluster association accuracy, we propose a new metric, Clustering Association Metric (CAM). For simplicity, we consider the case where there are only two domains. Assume that we discover \( h \) pairs of cluster associations \( \{ c_j, c'_j \}, 1 \leq j \leq h \), where \( c_j \) is a cluster in domain \( D_1 \) and \( c'_j \) is its corresponding cluster in domain \( D_2 \). Also assume that the ground-truth contains \( h \) pairs of cluster associations \( \{ t_i, t'_i \}, 1 \leq i \leq h \), where \( t_i \) is a cluster in domain \( D_1 \) and \( t'_i \) is its corresponding cluster in domain \( D_2 \). Then the CAM is defined by the following equation.
\[ CAM = \frac{1}{h} \sum_{i=1}^{h} \max_{1 \leq j \leq h} \frac{|(t_i \cup t'_i) \cap (c_j \cup c'_j)|}{|t_i \cup t'_i| \cap (c_j \cup c'_j)|} \]

where \( \cup \) is the set union and \( \cap \) is the set intersection. Here, in order to measure the similarity between an inferred cluster association \( \{ c_j, c'_j \} \) and the ground-truth \( \{ t_i, t'_i \} \), we use \[ \frac{|(t_i \cup t'_i) \cap (c_j \cup c'_j)|}{|t_i \cup t'_i| \cap (c_j \cup c'_j)|} \] to evaluate the degree of overlap between \( t_i \cup t'_i \) and \( c_j \cup c'_j \). For each ground-truth association pair \( \{ t_i, t'_i \} \), we get the maximal value of \[ \frac{|(t_i \cup t'_i) \cap (c_j \cup c'_j)|}{|t_i \cup t'_i| \cap (c_j \cup c'_j)|} \] for \( 1 \leq j \leq h \). The CAM is the average of the maximal values for all ground-truth pairs.

B. Simulation Study

We constructed a simulation study on the two synthetic networks \( G_1, G_2 \) in Fig. 1. We compare our method MCA with DNMTF, NMTF_Chris and RCC with respect to robustness to varying levels of noise in the cross-domain instance-level relationship matrix \( W_{12} \). Fig. 2 shows that MCA achieves...
much higher clustering accuracy than all existing methods at all noise levels. Fig. 3 demonstrates the clear advantage of MCA over existing methods in capturing cross-domain cluster associations.

C. DBLP Dataset

We also evaluate our method MCA using a labeled DBLP dataset [18], [19]. The dataset consists of papers and authors from 4 research areas: Database (DB), Artificial Intelligence (AI), Data Mining (DM) and Information Retrieval (IR). It contains 20 conferences and 4057 authors. These conferences are listed by area in Table III and the author distribution by area is shown in Table IV. We use $D_1$ to denote the author domain and $D_2$ to denote the conference domain. In $D_1$, the network $G_1$ represents the co-authorship. Each entry $A_{1ij}$ in the affinity matrix $A_1$ of $G_1$ is the number of papers coauthored by the $i$-th and $j$-th authors. The affinity matrix $A_2$ of $G_2$ represents the similarities between the topics covered by two conferences. To compute it, we first construct the term-conference matrix $F$, in which each entry $F_{ij}$ is the number of occurrences of the $i$-th term in the titles of papers published in the $j$-th conference. Thus each column $F_j$ of the matrix can be viewed as a feature vector describing the $j$-th conference. The similarity score of two conferences $j$ and $j'$ can be computed as $\|F_j - F_j'\|$. $W_{12}$ represents the relationships between authors and conferences, in which each entry denotes the number of papers that an author published in a given conference. A snapshot of the DBLP network used in our experiment can be seen in Figure 6.

To compare the robustness of different methods, we introduce noise by randomly shuffling a certain percentage of the entries in $W_{12}$. Fig. 4 shows that noise in the prior knowledge on cross-domain relationships does not affect the clustering accuracy of MCA in the conference domain at all, and only lowers the accuracy of MCA in the author domain modestly when $W_{12}$ is dominated by noise. Fig. 5 shows that the accuracy of the inferred cross domain cluster associations also only drops modestly for MCA when the noise level is very high. In contrast, we observe that all other methods are far more sensitive to noise, among which NMTF_Chris performs noticeably better than ONMTF and RCC.

To better understand how these methods perform, we list the top 4 associations between conference clusters and author clusters returned by MCA and NMTF_Chris in Table V when the noise level is set to 30%. We do not list the results by DNMTF and RCC because they return only a single cluster that includes all conferences in the conference domain. This is obviously not what we desire. From Table V, we observe that

![Figure 2](image-url)

**Fig. 2.** Clustering accuracy as a function of increasing percentage of noise in $W_{12}$ on simulated data.

![Figure 3](image-url)

**Fig. 3.** Cluster association accuracy as a function of increasing percentage of noise in $W_{12}$ on simulated data.

![Figure 4](image-url)

**Fig. 4.** Normalized Mutual Information with respect to different noise levels on the DBLP dataset.

![Figure 5](image-url)

**Fig. 5.** Cluster association accuracy with respect to different noise levels on the DBLP dataset.

| Number of Authors from Each Research Area in the DBLP Dataset |
|---------------------|-----|-----|-----|-----|
| DB | AI | DM | IR |
| 1197 | 1109 | 743 | 1006 |
| Percentage | 29.5% | 27.3% | 18.4% | 24.8% |

**TABLE IV.** Number of Authors from Each Research Area in the DBLP Dataset

**TABLE III.** List of Conferences from Each Research Area in the DBLP Dataset

<table>
<thead>
<tr>
<th>DB</th>
<th>AI</th>
<th>DM</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PODS</td>
<td>AAAI</td>
<td>ICML</td>
<td>SIGIR</td>
</tr>
<tr>
<td>SIGMOD</td>
<td>ICMI</td>
<td>ICDM</td>
<td>WWW</td>
</tr>
<tr>
<td>VLDB</td>
<td>IJCAI</td>
<td>SDM</td>
<td>WSDM</td>
</tr>
<tr>
<td>EDBT</td>
<td>CVPR</td>
<td>PKDD</td>
<td>ECIR</td>
</tr>
<tr>
<td>ICDE</td>
<td>ICML</td>
<td>PAKDD</td>
<td>CIKM</td>
</tr>
</tbody>
</table>
MCA produces the correct clustering result in the conference domain. The conference cluster in each of the top 4 pairs corresponds to a distinct research area. However, NMTF_Chris makes many mistakes. It splits the conferences from the Database area into two clusters. The third and fourth conference clusters are mixtures of conferences from different areas. In the author domain, for each author cluster, the percentage of authors from each of the 4 research areas is also shown in Table V. Each author cluster returned by MCA is primarily dominated by authors from one research area, as indicated by the largest percentage highlighted in bold in each column, which perfectly matches the area suggested by the associated conference cluster. For example, consider the 1st pair of conference cluster and author cluster returned by MCA. The conference cluster includes PODS, SIGMOD, VLDB, ICDE and EDBT, all of which are Database conferences. 94.2% of authors in the author cluster also come from the Database area. MCA correctly infers this association between the author cluster and conference cluster. We can make same observation on the remaining cluster pairs by MCA in Table V. It demonstrates that MCA can discover meaningful associations between clusters from different domains. From Table V, we also observe that some conference clusters and author clusters discovered by NMTF_Chris represent a mixture of multiple research areas. To further quantify this observation, for the 4 author clusters, we compute the KL-divergence between the research area distributions of each pair of author clusters. A KL-divergence of 1 indicates that authors in the two clusters are from two distinct areas. A KL-divergence of 0 indicates that the two author clusters have identical research area distributions and thus are not distinguishable from each other. We use dark color for small KL-divergence and light color for large KL-divergence in Fig. 7. A diagonal entry depicts the KL-divergence of an author cluster to itself which is always 0. Off-diagonal entries correspond to KL-divergence of two author clusters — the larger the KL-divergence, the better the clustering result. We observe that the 4 clusters by MCA all have large KL-divergence to each other but the first two clusters by NMTF_Chris have small KL-divergence. This proves that MCA is more robust to random noise.

Duplicate Names: It is possible that some authors may have the same name in the DBLP database. Publications by these authors might be mistakenly associated with other authors with the same name. In order to evaluate the robustness of our method in this context, we first randomly pick a certain percentage of authors and then randomly pair them up. We "pretend" that the two authors in each pair use the same name and thus their publications are not distinguishable. We replace the two corresponding row vectors in $W_{12}$ by their average vector. Since only a small percentage of authors may have this issue, we only test the robustness for up to 30% of duplicate names. Fig. 8 shows that MCA can successfully resolve the ambiguity and thus its clustering accuracies and cluster association accuracy are not impacted. NMTF_Chris is the second best, having comparable accuracies in cluster association and author clustering, but failing short on the conference clustering accuracy.

D. Yeast eQTL Dataset

Expression quantitative trait loci (eQTL) mapping is the process of identifying single nucleotide polymorphisms (SNPs) that play important role in the expression of genes. It has
been widely used to dissect genetic basis of complex traits [5].
Traditionally, associations between individual expression traits
and SNPs are assessed separately [20]. Since genes in the same
biological pathways are often coregulated and may share a
common genetic basis [21], it is crucial to understand how
multiple modestly-associated SNPs interact to influence the
phenotypes [22]. To answer this question, several approaches
have been proposed to study the joint effect of multiple SNPs
by testing the association between a set of SNPs and a gene
expression trait [23]–[26]. Despite their success, these methods
have two common limitations. First, only the association
between a set of SNPs and a single expression trait is studied.
Therefore, they overlook the joint effect of a set of SNPs on
the activities of a set of genes, which may act and interact with
each other to achieve certain biological function. Second, the
SNP sets used in these methods are usually taken from known
biological pathways, which are far from being complete.
These methods cannot identify unknown associations between
SNP sets or gene sets. To better elucidate the genetic basis
of gene expression and understand the underlying biology
pathways, it is highly desirable to develop methods that can
automatically infer association between a group of SNPs and
a group of genes. The process of identifying such associations
is referred to as group-wise eQTL mapping, to distinguish it
from the individual eQTL mapping [6] process that identifies
associations between individual SNPs and genes. The MCA
method proposed in this paper is suitable for group-wise eQTL
mapping.

We compare MCA with NMTF_Chris, DNMTF and RCC
on a yeast eQTL dataset [27]. This dataset originally includes
expression profiles of 6229 genes and genotype profiles of
2956 SNPs. After preprocessing (e.g., removing missing val-
ues), the dataset is reduced to 1017 SNPs and 4474 genes
expression profiles.

We denote the SNP domain as $\mathcal{D}_1$ and the gene domain as
$\mathcal{D}_2$, respectively. The SNP interaction network $\mathcal{G}_1$ is generated
as in [28]. The gene interaction network $\mathcal{G}_2$ is constructed by
computing the Pearson’s correlation of the expression levels
of each pair of genes.

$$r_{XY} = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2} \sqrt{\sum_i (Y_i - \bar{Y})^2}}$$ (24)

where $\mathbf{X}$ and $\mathbf{Y}$ are vectors representing the expression profiles
of the two genes. $X_i$ and $Y_i$ are the $i$th components of $\mathbf{X}$
and $\mathbf{Y}$, respectively. From Eq. (24), the value of Pearson’s
correlation $r$ ranges from $-1$ to $1$, where $1$ means that two genes are completely positively correlated and $-1$ means that they are completely negatively correlated. The edge between genes $X$ and $Y$ in the gene interaction network is weighted by $\mid r_{XY}\mid$. The association matrix $W$, is given by the association tests between individual SNPs and individual genes using PLINK [29].

**Gene Ontology Enrichment Analysis:** Since there is no ground-truth in the Yeast eQTL dataset, we cannot measure the clustering accuracy and cluster association accuracy directly. Here, we evaluate the quality of our result by the Gene Ontology Enrichment Analysis (GOEA) [30]. For each inferred gene cluster $c_i$, we identify the most significantly enriched Gene Ontology categories [31]. The significance ($p$-value) is determined by the Fisher’s exact test. The raw $p$-values are further calibrated to correct for the multiple testing problem [32]. To compute calibrated $p$-values for $c_i$, we perform a randomization test, wherein we apply the same test to $1000$ randomly created gene sets that have the same number of genes as in $c_i$. In order to evaluate the clusters in the SNP domain, we first need to map the SNPs in a cluster to their nearest genes on the genome, and then apply the standard procedure of GOEA on the set of genes to compute a $p$-value. In Fig. 9, clusters are arranged in ascending order of their $p$-values. We consider the clusters with $p$-value less than 0.05 to be significant. The numbers of significant gene and SNP clusters are listed in Table VI. Not surprisingly, MCA can identify more significant clusters in both gene and SNP domains than the competitors.

**TABLE VI. THE NUMBER OF SIGNIFICANTLY ENRICHED CLUSTERS MEASURED BY GOEA**

<table>
<thead>
<tr>
<th></th>
<th>MCA</th>
<th>NMTF_Chris</th>
<th>DNMTF</th>
<th>RCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>gene</td>
<td>36</td>
<td>34</td>
<td>25</td>
<td>20</td>
</tr>
<tr>
<td>SNP</td>
<td>99</td>
<td>98</td>
<td>82</td>
<td>62</td>
</tr>
</tbody>
</table>

**E. Gene Disease Dataset**

We further evaluate our algorithm MCA on a gene disease network dataset [33]. The dataset contains 590 disease phenotypes in 20 disease classes and 7997 genes in 200 gene pathways. There are two domains: gene domain $D_1$ and disease phenotype domain $D_2$. $G_1$ represents the "functional" relationships between genes which are measured by interactions between the proteins transcribed from the genes, because most genes "perform" their functions through their transcribed proteins. This protein-protein interaction network can be obtained from HPRD [34]. The relationships among phenotypes are represented by a phenotype similarity network $G_2$, which is obtained from [35]. It is an undirected network with vertices representing OMIM [36] disease phenotypes and edges (with weights between 0 and 1) representing the similarities between phenotypes measured by their co-occurrences in clinical synopsis records. The associations between disease phenotypes and genes are also available in OMIM. We evaluate the clustering accuracy in each domain using the normalized MI discussed in Section IV-A. The first row in Table VII is the normalized MI in the phenotype domain and the second row is the normalized MI in the gene domain. As we can see, MCA is again the winner.

**F. Performance Evaluation**

In this section, we study the run-time performance of MCA, measured by the number of iterations before converging to a local optima. Table VIII summarizes the network size and the number of iterations upon convergence on difference data sets. We observe that MCA can converge within a reasonable number of iterations even for large networks. As expected, the number of iterations will increase as the network size increases. Usually, several hundreds of iterations are needed before convergence, but the actual running time is fast. Table IX shows the time used by different methods to convergence on the DBLP dataset. All methods except RCC run very fast. We can conclude that MCA can achieve much better accuracy without entailing more computation time.

**TABLE VIII. NUMBER OF ITERATIONS TO CONVERGE**

<table>
<thead>
<tr>
<th>dataset</th>
<th>size of network $\mathcal{D}_1$</th>
<th>size of network $\mathcal{D}_2$</th>
<th>number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>synthetic</td>
<td>160</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>DBLP</td>
<td>441</td>
<td>20</td>
<td>80</td>
</tr>
<tr>
<td>eQTL</td>
<td>1017</td>
<td>4474</td>
<td>741</td>
</tr>
<tr>
<td>Gene-disease</td>
<td>3619</td>
<td>966</td>
<td>144</td>
</tr>
</tbody>
</table>

**TABLE VII. RESULTS ON THE GENE DISEASE NETWORK**

<table>
<thead>
<tr>
<th>method</th>
<th>MCA</th>
<th>NMTF_Chris</th>
<th>DNMTF</th>
<th>RCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized MI phenotype</td>
<td>0.19</td>
<td>0.14</td>
<td>0.14</td>
<td>0.16</td>
</tr>
<tr>
<td>Normalized MI gene</td>
<td>0.05</td>
<td>0.02</td>
<td>0.05</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Fig. 9. Gene ontology enrichment analysis on the yeast eQTL data.
V. CONCLUSION

In this paper, we propose a novel algorithm, MCA, for network clustering across multiple related domains. By leveraging the duality between single network clustering and inferring cross-network cluster alignment, MCA well incorporates any prior knowledge on cross-network instance relationships into multi-network clustering. The algorithm is robust to noise and is capable of detecting cross-domain associations between clusters, which was never addressed in previous study. Extensive experiments on both synthetic and several real datasets demonstrate the effectiveness and efficiency of MCA and its advantages over existing methods.

VI. ACKNOWLEDGMENT

This work was partially supported by the National Science Foundation grants IIS-1162374, IIS-1218036, IIS-1313606 and IIS-1017415, by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, by National Institutes of Health under the grant number R01LM011986 and U54GM114833-01, Region II University Transportation Center under the project number 49997-33 25.

REFERENCES