Community-Enhanced Semi-seeded Network Alignment (CESSNA): A Robust Method with Application to Microbiome Networks

Sijing Yu
Department of Computer Science
University of Maryland
College Park, MD, USA

Daniel L. Sussman
Department of Mathematics
and Statistics
Boston University
Boston, MA, USA

Vince Lyzinski
Department of Mathematics
University of Maryland
College Park, MD, USA

Abstract

Network alignment (i.e., graph matching) is a fundamental, though computationally challenging, problem that seeks to identify correspondences between vertices in network data. We present Community-Enhanced Semi-seeded Network Alignment (CESSNA), a novel algorithm that integrates community structure and partial seed information to improve matching efficiency and accuracy. CESSNA decomposes the global matching problem into smaller, community-based blocks, enabling efficient block-wise gradient descent and reducing computational complexity to that of the largest non-seeded community. The method flexibly incorporates both true and inferred communities, maintaining robustness even in the presence of noise and limited seed data. Experiments on synthetic and real-world datasets demonstrate that CESSNA consistently outperforms traditional seeded and unseeded graph matching approaches, achieving up to a 46-fold increase in accuracy over state-of-the-art methods without any seeds on the Wikipedia dataset. Furthermore, we present an innovative application of CESSNA to microbiome data, showing the capacity of this approach for robust, multiscale graph comparison in complex network data. These findings highlight the potential of CESSNA for addressing a broad spectrum of non-traditional network alignment problems, and emphasize CESSNA's ability to accomplish efficient and accurate network alignment and its utility for uncovering new insights in biologically hierarchical data.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

1 INTRODUCTION

Consider two graphs G_1 and G_2 with respective adjacency matrices A and B. The network alignment problem, also known as the *graph matching problem (GMP)*, aims to find the permutation matrix relabeling the vertices of B that minimizes the induced structural difference between the two graphs. The GMP is a challenging problem in computer science with applications in diverse areas including pattern recognition (Conte et al., 2004; Lee et al., 2010), computer vision (Escolano et al., 2011; Lin et al., 2010), and biology (Chen et al., 2015). While the related *graph isomorphism* problem—determine the existence of a permutation matrix P such that $A = PBP^\top$—has been shown to have at worst quasipolynomial complexity (Babai, 2016), the GMP in its most general form is equivalent to the classic *quadratic assignment problem (QAP)*, which is NP-hard (Sahni and Gonzalez, 1976). Given the hardness of exact GMP, much research has focused on efficient approximations. Spectral methods frame GMP as a spectral alignment problem (Bai et al., 2004), combinatorial algorithms incrementally build matchings from seed pairs (Yartseva and Grossglauser, 2013a; Narayanan and Shmatikov, 2009), and relaxation-based approaches solve approximate QAPs using continuous optimization frameworks (Vogelstein et al., 2015). Here, we address an approximate, relaxed form of the original problem.

We propose a novel GMP algorithm that incorporates partially observable community structures without relying on strict assumptions about seed nodes or latent correspondences. When seed information is available, the algorithm can readily incorporate it to further improve performance. The scenario of partially observ-

able communities is natural in many theoretical models and real-world datasets. For example, the widely used stochastic block model (SBM) intrinsically defines communities, and under mild assumptions the ground-truth clustering can be recovered exactly with high probability (e.g., through semidefinite programming (Lim et al., 2014)). Moreover, community structure is a prevalent feature in real-world networks including social networks (Girvan and Newman, 2002) and biological networks (Newman, 2006). In situations where the true underlying clusters are not fully known, numerous community detection algorithms have been proposed, validated, and demonstrated to perform effectively (Tokala et al., 2025).

Our novel algorithm builds upon (Fishkind et al., 2019) to incorporate community information into network alignment while maintaining adaptability to settings with and without seed nodes. Specifically, we introduce Community-Enhanced Semi-seeded Network Alignment (`CESSNA`), effectively leveraging varying amounts of community data, and offer insights into its performances. Our main contributions are threefold:

1: We propose `CESSNA`, an optimization framework with proven theoretical guarantees for the graph matching problem that naturally integrates community information including ground-truth communities or solely results of inferred clusters, and demonstrates its robustness to noise. In addition, we offer novel empirical insights into `CESSNA` matching results from an information theory perspective.

2: We adapt `CESSNA` to multiscale graph matching for data with hierarchical community structures, enabling novel insights into biological networks through a case study on a recent sequencing expression dataset.

3: We provide a computationally efficient implementation with seamless partial seed integration and show its effectiveness compared to other graph matching algorithms, enhancing performance while maintaining algorithmic simplicity.

2 THE GRAPH MATCHING PROBLEM

For integer n we define $[n] := \{1, 2, \ldots, n\}$. Given adjacency matrices $A, B \in \mathbb{R}^{n\times n}$, the Graph Matching Problem (GMP) seeks to find

$$\operatorname*{argmin}_{P\in\mathcal{P}_n} \left\|A - PBP^\top\right\|_F^2 = \operatorname*{argmax}_{P\in\mathcal{P}_n} \operatorname{tr}(APB^\top P^\top), \quad (1)$$

where $\|\cdot\|_F^2$ is the Frobenius norm, and $\mathcal{P}_n$ is the set of $n \times n$ permutation matrices. Dropping constant terms of the expanded objective gives the right term above. Note that an optimal solution representing an exact isomorphism may not exist for certain graph pairs.

The GMP in Eq. 1 is combinatorially challenging, and it is common to relax $\mathcal{P}_n$ to its convex hull, the set of doubly stochastic matrices $\mathcal{D}_n$; (Lyzinski et al., 2016) formalized the *indefinite* relaxation of Eq. 1 as

$$\operatorname*{argmin}_{D\in\mathcal{D}_n} -\operatorname{tr}(ADB^\top D^\top), \quad (2)$$

in contrast to the convex relaxation minimizing $\|A - DBD^\top\|_F^2$. While the convex relaxation has been used extensively (Aflalo et al., 2015; Ding and Wolkowicz, 2009), (Lyzinski et al., 2016) proves that under mild model assumptions the indefinite relaxation approximates the optimal permutation better than the convex relaxation; this is further explored in (Maron and Lipman, 2018). While generally indefinite quadratic programs including the one generated from above are NP-hard, (Vogelstein et al., 2015) proposes to locally optimize the indefinite relaxation with the Frank-Wolfe (FW) algorithm (Frank et al., 1956), outperforming previous convex relaxation methods (Zaslavskiy et al., 2009). Further work in (Maron and Lipman, 2018) shows that under certain conditions, the indefinite relaxation becomes concave, offering the benefit that all local minima are permutations.

Motivated by common real-world applications where a partial correspondence is known across A and B (Narayanan and Shmatikov, 2009), the *seeded graph matching problem (SGMP)* uses *seeds*—vertices whose correspondence is available a priori—to more efficiently align a pair of graphs (Fishkind et al., 2019). The SGMP is formalized as $\operatorname{argmin}_{P\in\mathcal{P}_n} \left\|A - (I_m \oplus P)B(I_m \oplus P)^\top\right\|_F^2$ where $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ and $B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$, $A_{11}, B_{11} \in \mathbb{R}^{m\times m}$, $A_{12}, B_{12} \in \mathbb{R}^{m\times n}$, $A_{21}, B_{21} \in \mathbb{R}^{n\times m}$, and $A_{22}, B_{22} \in \mathbb{R}^{(n-m)\times(n-m)}$, I_m is the m-by-m identity matrix denoting seed information, and $\oplus$ is the direct sum of matrices. Here, w.l.o.g. we assume the seeds are vertices $[m]$ in both graphs.

Although SGMP has better matching results than the original GMP (Lyzinski et al., 2014; Mossel and Xu, 2020; Kazemi et al., 2015), its reliance on seeds imposes significant limitations on its practical applicability; indeed, seeded vertices may be expensive to obtain in practice. Various approaches to SGMP have been proposed (e.g., the percolation approach of (Narayanan and Shmatikov, 2009), and the neighborhood alignment of (Mossel and Xu, 2020)), though these approaches inherently require a minimum seed set size for successful alignment (Yartseva and Grossglauser, 2013b). Moreover, performance can be sensitive to algorithmic parameters (Kazemi et al., 2015; Yu et al., 2021). SGMP is further complicated by the

lack of correctness guarantees of seed nodes especially when seeds are algorithmically generated instead of ground-truth (Zhu et al., 2011; Zhou et al., 2017) Additionally, obtaining ground-truth seeds often requires labor-intensive processes in practice (Patsolic, 2020). Our work tackles these challenges by achieving comparable performances without hard seeds, while retaining the flexibility to leverage available seed information.

While exact seed information is often limited, extensive research has demonstrated the prevalence of community structures across diverse networks (Girvan and Newman, 2002; Fortunato and Castellano, 2007; Lancichinetti et al., 2011). Robust graph clustering algorithms—such as modularity optimization (Newman, 2006; Blondel et al., 2008) and model-based approaches (Li et al., 2007)—reliably detect communities in networks without ground-truth labels. As traditional graph data represent pre-existing structures, graphs exhibiting community structure can also be generated from prior information—such as biological classification systems—with widely useful applications (Sah et al., 2014; Mazein et al., 2024). For instance, in taxonomic hierarchies (phylum, class, family, genus, species), co-occurrence matrices derived from feature vectors naturally form hierarchical communities. These structures emerge because closely related taxa (e.g. species within a genus) share more co-occurrence patterns than distant ones. While related problems receive wide attention in biological research, existing methodologies exhibit limitations including not making use of the network structure (Bartomeus et al., 2016), or stringent assumptions about statistical models (Legras et al., 2019). In Section 5, we will demonstrate `CESSNA`'s capacity to use multiscale graph matching with the hierarchical community structures to uncover latent relationships between distinct sample groups using co-occurrence matrices.

3 METHOD AND THEORETICAL ANALYSIS

As described previously, our method tackles the indefinite graph matching relaxation Eq. 2, before projecting the solution back onto $\mathcal{P}_n$. Building on the FAQ algorithm (Vogelstein et al., 2015), which employs FW optimization (Frank et al., 1956), we develop a community-enhanced procedure. Consider a partition of the vertices of A and B into *the same* K communities (with community i of size n_i). Letting $i \in [K] = \{1, 2, \ldots, K\}$, we posit that the clusters corresponding to $A_{i,i} \in \mathcal{R}^{n_i \times n_i}$ and $B_{i,i} \in \mathcal{R}^{n_i \times n_i}$ are matchable across graphs. Solving Eq. 2 with this added restriction is equivalent to restricting $\mathcal{P}_n$ to block-diagonal P of the form $P = P_1 \oplus P_2 \oplus \cdots \oplus P_K$, with $P_i \in \mathcal{P}_{n_i}$. With this, Eq. 2 decomposes as (where $A_{i,j}$ is the subgraph of A composed of edges across communities i and j; similarly for $B_{i,j}$)

$$-\operatorname{tr}(A^\top P B P^\top) = -\sum_{i,j} \operatorname{tr}(A_{i,j}^\top P_i B_{i,j} P_j^\top) \quad (3)$$

solving this problem suggests a block-gradient descent adaptation of the FAQ (Vogelstein et al., 2015), which we implement as follows.

Algorithm 1 Community-Enhanced Graph Matching

Input: Adjacency matrices $A, B \in \{0,1\}^{n \times n}$; partition sizes $\vec{n} = (n_1, n_2, \ldots, n_K)$ and cluster–to–cluster matching $\phi : [K] \mapsto [K]$ (without loss of generality, let $\phi = \mathrm{id}_K$); tolerance $\varepsilon > 0$;
Step 1. Initialize $P^{(0)}$;
while $\|P^{(t)} - P^{(t-1)}\|_F > \varepsilon$ **do**
  Step 2. Let $\sigma \in S_K$ be a uniformly random permutation of $[K]$;
  for $\ell = 1$ **to** K **do**
    i. Set $j = \sigma(\ell)$;
    ii. Compute the j-th block gradient via Eq. (4)
    iii. Compute the j-th block search direction, $Q_j^{(t)}$, via Eq. (5)
    iv. Compute the step size, β_j, in the direction of $Q_j^{(t)}$ via Eq. (6)
    v. Set $P_j^{(t+1)} = \beta_j P_j^{(t)} + (1 - \beta_j) Q_j^{(t)}$;
  end for
end while
for $j = 1$ **to** K **do**
  i. Project $P_j = \max_{Q_j \in \mathcal{P}_{n_j}} \operatorname{tr}\left(Q_j^\top P_j^{(\mathrm{final})}\right)$;
end for
Output: Permutation matrix $P = P_1 \oplus P_2 \oplus \cdots \oplus P_K$ matching graphs A and B.

3.1 Community-Enhanced Graph Matching

The Community-Enhanced Graph Matching algorithm building on FAQ proceeds via:

Step 1: Initialize the matching at $P^{(0)}$. We can initialize at the flat barycenter $P_i^{(0)} = \mathbf{1}\mathbf{1}^\top / n_i \in \mathbb{R}^{n_i \times n_i}$ (where $\mathbf{1}$ is the length $n_i - m_i$ vector of all 1's), though (Lyzinski et al., 2016) noted that performance can often be augmented by initializing at the solution of the convex graph matching relaxation $P_i^{(0)} \in \operatorname{argmin}_{D \in \mathcal{D}(n_i)} \|A_{i,i} D - D B_{i,i}\|_F^2$.

Step 2: While $\|P^{(t+1)} - P^{(t)}\|_F > \varepsilon$ for a preset threshold ε, apply the following block-FW adaptation. If at iteration t, update $P^{(t)}$ as follows. Let $\sigma \in S_K$ be a uniformly random permutation of $[K]$. For $\ell = 1, 2, \ldots, K$, set $j = \sigma(\ell)$ and update as follows:

i. Compute the j-th block gradient of Eq.;

$$\nabla_j = -\sum_{i \neq j} A_{j,i} P_i^{(t)} B_{j,i}^T - \sum_{i \neq j} A_{i,j}^T P_i^{(t)} B_{i,j} - A_{j,j} P_j^{(t)} B_{j,j}^T - A_{j,j}^T P_j^{(t)} B_{j,j}; \quad (4)$$

Algorithm 2 Community-Enhanced Semi-seeded Network Alignment (CESSNA)

Input: same as Algorithm 1
Adapted Step 1: Let $P_i^{(0)} = I_{m_i} \oplus \frac{1}{n_i - m_i}\mathbf{1}\mathbf{1}^\top$ or convex matching
while $\|P^{(t)} - P^{(t-1)}\|_F > \varepsilon$ **do**
  Adapted Step 2. Perform same calculations as **step 2** in Algorithm 1 while preserving constant terms given by the seed information throughout to avoid repetitive computation: Let $\sigma \in S_K$ be a uniformly random permutation of $[K]$;
  for $\ell = 1$ **to** K **do**
    i., iii., iv., and v., Same as in Algorithm 1;
    Modified ii. Compute the j-th block gradient via Eq. (8)
  end for
end while
Output: same as Algorithm 1

ii. Compute the j-th block search direction

$$Q_j^{(t)} = \min_{Q \in D(n_j)} \operatorname{tr}\left[\nabla_j^T Q\right]; \tag{5}$$

iii. Compute the optimal step size in the direction of $Q_j^{(t)}$. With $P_{j,\beta}^{(t)} = \beta P^{(t)} + (1-\beta)P_{Qj}^{(t)}$ with $P_{Qj}^{(t)} = \left(P_1^{(t)} \oplus \cdots \oplus P_{j-1}^{(t)} \oplus Q_j^{(t)} \oplus P_{j+1}^{(t)} \oplus \cdots \oplus P_K^{(t)}\right)$ we set the optimal steps-size equal to

$$\beta_j = \min_{\beta \in [0,1]} \operatorname{tr}\left(A P_{j,\beta}^{(t)} B^T (P_{j,\beta}^{(t)})^T\right); \tag{6}$$

iv. Set $P_j^{(t+1)} = \beta_j P_j^{(t)} + (1-\beta_j) Q_j^{(t)}$;

Step 3: Project the final iterate, $P^{(\text{final})} = P_1^{(\text{final})} \oplus P_2^{(\text{final})} \oplus \cdots \oplus P_K^{(\text{final})}$ onto $\mathcal{P}_n$ by solving the linear assignment problem $P = \max_{Q \in \mathcal{P}_n} \operatorname{tr}(Q^\top P^{(\text{final})})$ using Jonker-Volgenant (JV) algorithm (Jonker and Volgenant, 1988); this can be parallelized by separately projecting $P_j = \max_{Q_j \in \mathcal{P}_{n_j}} \operatorname{tr}(Q_j^\top P_j^{(\text{final})})$.

3.2 CESSNA

Given above Community-Enhanced Graph Matching Algorithm 1, adding seeds as in (Fishkind et al., 2019) is immediate. In fact, seeding Algorithm 1 amounts simply to seeding the gradient calculation and search direction calculation in **Step 2** above. To illustrate, in addition to restricting $\mathcal{P}_n$ to P of the correct block form, we further restrict (assuming, w.l.o.g., the first m_i seeds in community i are seeded across graphs)

$$P_i = \begin{bmatrix} P_{i_{11}} & P_{i_{12}} \\ P_{i_{21}} & P_{i_{22}} \end{bmatrix} = \begin{bmatrix} I_{m_i} & 0 \\ 0 & P_{i_{22}} \end{bmatrix}, \; P_{i_{22}} \in \mathbb{R}^{(n_i - m_i)\times(n_i - m_i)}$$

the block I_{m_i}—the m_i-by-m_i identity matrix—denotes the seed information for the i-th community. Therefore, from Algorithm 1 we have **Adapted Step 1** (see Algorithm 2) as above. In addition, the indefinite relaxation in Eq. 3 can be further decomposed and simplified as

$$-\sum_{i,j}\Big[\operatorname{tr}(A_{ij_{11}}^\top B_{ij_{11}}) + \operatorname{tr}(A_{ij_{21}}^\top P_{i_{22}} B_{ij_{21}}) + \operatorname{tr}(A_{ij_{12}}^\top B_{ij_{12}} P_{j_{22}}^\top) + \operatorname{tr}(A_{ij_{22}}^\top P_{i_{22}} B_{ij_{22}} P_{j_{22}}^\top)\Big]$$

with $A_{ij} = \begin{bmatrix} A_{ij_{11}} & A_{ij_{12}} \\ A_{ij_{21}} & A_{ij_{22}} \end{bmatrix}$, $B_{ij} = \begin{bmatrix} B_{ij_{11}} & B_{ij_{12}} \\ B_{ij_{21}} & B_{ij_{22}} \end{bmatrix}$ with P_i as above; note that any terms involving both $P_{i_{11}} = I_{m_i}$ and $P_{j_{11}} = I_{m_j}$ remain constant during calculation. Therefore, by saving the constant terms from the seed information, the computational complexity can be significantly reduced, leading to a modified equation calculated in **ii.** in **Adapted Step 2**:

$$-\nabla_j \sum_{i,j} \operatorname{tr}(A_{i,j}^\top P_i B_{i,j} P_j^\top) = -\sum_{ij}(A_{ij_{12}}^\top B_{ij_{12}} \tag{7}$$

$$+ B_{ij_{21}} A_{ij_{21}}^\top + A_{ij_{22}}^\top P_{i_{22}} B_{ij_{22}} + A_{ij_{22}} P_{i_{22}} B_{ij_{22}}^\top) \tag{8}$$

Similar techniques can also be applied to **iv.** where the calculation involves P_i. Consequently, although the overall complexity is bottlenecked by $O(Kn^3)$ from the linear assignment problem solver (LAP) (Jonker and Volgenant, 1988), we have

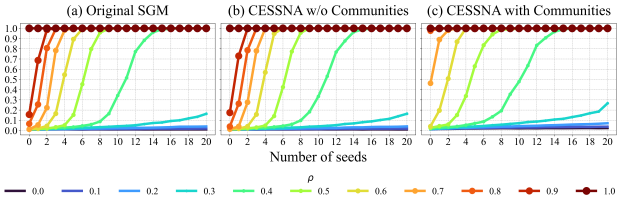
Proposition 3.1. *Given bounded iterates, CESSNA has complexity of $O(Kn_i^3)$ where $n_i \le n$ is the size of the greatest non-seeded part from all communities.*

Notice that since we only need to update non-seeded information for each community and we calculate with respect to each community in **Step 2**, we reduce the largest input size of the linear assignment problem to the size of largest non-seeded part from any community. This improvement can lead to significant reduction in time/space in practice. Lastly, observe that although CESSNA is not immediately parallelizable, it allows the employment of asynchronous parallelization in a relatively straightforward way similar to existing works (Liu et al., 2014; Richtárik and Takáč, 2016). In addition, CESSNA has strong convergence properties both theoretically and empirically; see Section 3.3.

3.3 Theoretical/Empirical Convergence

Introduced by (Holland et al., 1983), the stochastic block model (SBM) is a widely used model for graphs with latent community structure. Using correlation to endow a pair of SBM's with a latent alignment, this model provides an ideal setting for exploring CESSNA empirically and theoretically.

Definition 1 (Correlated Stochastic Blockmodel)**.** ***(CorrSBM)*** *Let $V = [n]$ be partitioned into K blocks $\{C_i\}_{i=1}^K$, with g the associated block membership func-*



× accuracy	$\rho = 0.0$	$\rho = 0.1$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$	$\rho = 1.0$
0 seed	1.51	1.69	2.54	39.43	6.39	1.0
1 seeds	1.70	1.76	3.02	16.35	1.46	1.0
5 seeds	1.96	1.95	4.13	1.00	1.00	1.0
10 seeds	2.06	2.02	1.00	1.00	1.00	1.0

Figure 1: Top: Matching Ratios w.r.t. Number of Seeds for various ρ on n=300 Vertices. Note when there is no community information, `CESSNA` is effectively equivalent to `SGM`. Bottom: Fold increase in accuracy compared to `SGM` algorithm (Fishkind et al., 2019)

tion. Given symmetric matrices $\mathbf{P} \in [0,1]^{K\times K}$ *(the block probabilities) and* $\mathbf{R} \in [0,1]^{K\times K}$ *(the block correlations), we say that* $(A,B) \sim \mathrm{CorrSBM}(\mathbf{P},\mathbf{R},g)$ *if:*

1. A, B *are marginally* $\mathrm{SBM}(\mathbf{P},g)$*; i.e.,* $\forall\ u < v$, $A[u,v] \overset{ind.}{\sim} \mathrm{Bern.}(P[g(u),g(v)])$ *(similarly for* B*).*
2. *The collection* $\{A[u,v], B[i,j]\}_{\{u,v\},\{i,j\}\in\binom{V}{2}}$ *are mutually independent except that* $\forall\{u,v\} \in \binom{V}{2}$, $\mathrm{corr}(A[u,v], B[u,v]) = R[g(u),g(v)]$.

In this model, we derive the following theoretical guarantees for our algorithm, which are heavily inspired by the analogous two step analysis of the classical FW-based matching provided in Fang et al. (2018). To ease exposition, we will consider the setting where all communities are the same order $c = |\{v \in V \text{ s.t. } g(v) = i\}|$, and where $K = n/c$, and where $\mathbf{P}$ and $\mathbf{R}$ are $\Theta(1)$ entrywise. The proof can be found in the supplement, as can an analogous one-step convergence result.

Theorem 1. *Suppose* $(A,B) \sim \mathrm{CorrSBM}(\mathbf{P},\mathbf{R},g)$. *Let* $\delta \in (0,1/2)$. *There exist constants* $a_1 > 0$ *and* $a_2 > 0$ *such that if* $\ell = a_1\sqrt{Kc^{1+2\delta}}$ *and* $m = a_2 c^{1-\delta}\log c/\sqrt{K}$ *satisfying* $m = \omega(\sqrt{c\log c})$; $m = o(c/(K\log^2 c))$, *and if for all* $i,j \in [K]$, *Suppose that for all* $i,j \in [K]$,

$$\mathbb{E}[\mathrm{tr}(A_{i,j}DB_{i,j}(Q-P))] \geq C\,\mathrm{tr}(D)(\mathrm{tr}(Q)-\mathrm{tr}(P))$$

for all permutation matrices D, P, Q *with* $\mathrm{trace}(P) \leq c-m$, $\mathrm{trace}(D) \geq \ell$, *and* $\mathrm{trace}(Q) \geq c-m$. *Then there exists a constant* $a_3 > 0$ *such that with probability at least* $1 - a_3K^5e^{-\Theta(c)}$, ***CESSNA*** *(given the true communities) will converge to the true correspondence,* I, *in at most two steps starting from any* $P = \oplus_i P_i$ *with all* $\ell \leq \mathrm{trace}(P_i) \leq c-2m$.

To explore the convergence guarantee above empirically, we run 10 runs with random seeds for each number of seeds for *Wikipedia* dataset (which will be introduced in detail in Section 4.2), and record the graph

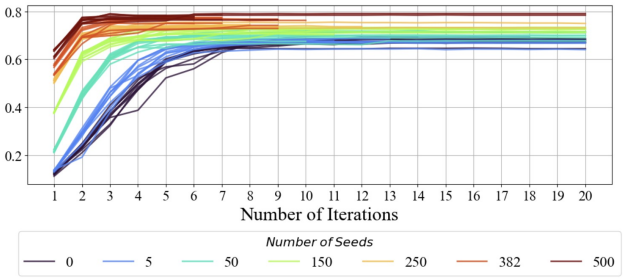


Figure 2: `CESSNA` Convergence results on *Wikipedia* data

matching accuracy with respect to the number of iterations. In Figure 2, we see: nearly all runs converge quickly (within 10 iterations); faster convergence correlates with better performance; and more seeds lead to faster convergence and better performance. This aligns with our result above where strong convergence guarantees are present for high-quality starting inputs. We also see that runs that take many iterations to converge lead to often substandard matchings; we could take advantage of this observation to further optimize our runtime by employing early stopping heuristics to abort longer, often suboptimal algorithm runs.

4 EXPERIMENTAL ANALYSIS

Ground-truth communities may not be available in many real-world situations, and various clustering algorithms have been proposed to detect communities (Newman, 2006; Blondel et al., 2008; Li et al., 2007). In such cases, where no ground-truth decomposition of P is available, we employ the efficient Leiden algorithm (Traag et al., 2019) to iteratively identify each community $P_i \in \mathcal{P}_{n_i}$ as input to our algorithm. Note that this is artificial, as we use the *same* community memberships across both graphs although the clustering is performed in a single graph; that said, our approach appears to be robust to a fair amount of clustering errors (see Figure 3). If we clustered each graph separately, we would need to align the clusters before proceeding introducing additional complexity, and we do not pursue this further here.

We experimentally evaluate `CESSNA` on both synthetic and real-world data. For synthetic data, we assess performance under assumptions about inter-graph correlations. For real-world data, we compare accuracies across scenarios utilizing partial, complete, and inferred communities. We implement our algorithm in Python and run experiments on Google Colab using TPU v6e and other available processors. [1]

4.1 Synthetic Data—generalized CorrSBM

Similar to the setting of (Fishkind et al., 2019), simulations are run on 150 independent pairs of $n = 300$

[1] Code is available at `github.com/SijingYu/CESSNA`

vertex graphs A and B for seed value $s \in \{1, 2, ..., 20\}$ for each $\rho \in \{0, 0.1, .., 0.9, 1\}$. Here A, B are drawn from the 3-block CorrSBM$(\mathbf{P}, \mathbf{R}, g)$ in a way that $n/3$ vertices are in each of the three blocks, and $\mathbf{R}$ is entrywise constant equal to ρ. We compared our results from CESSNA with the community information (Figure 1(c)) from SBM above with the version without community information (Figure 1(b)) and the SGM algorithm (Figure 1(a)). Note that due to implementation details (involving the LAP subroutine solver), we get slight improvements compared to the original SGM algorithm without community information. Figure 1 demonstrates that incorporating community information enables our algorithm to substantially reduce SGM's dependence on seed information. Notably, our method outperforms SGM by achieving a 39-fold and 6-fold increases in accuracy in the absence of any seed node for $\rho = 0.7$ and $\rho = 0.9$ resp. In addition, with 2 seeds and $\rho = 0.6$, CESSNA achieves a matching ratio well above 40%, an 11-fold increase in the accuracy of the original SGM. Moreover, CESSNA makes substantial improvements as the number of seeds increases.

4.2 Real data: C.elegans, Wikipedia

C.elegans Mapped by (Varshney et al., 2011) and representing a roundworm with 279 vertices as its neurons and edges as synapses, the C.elegans graphs have been studied extensively (Lall et al., 2006; Singh et al., 2008; Valouev et al., 2008). Here, A_{gap} and B_{chem} denote the gap junctional and chemical connectomes, respectively. We use the symmetric adjacency matrix of A_{gap} to generate communities using the Leiden algorithm (Traag et al., 2019); these communities are artificial (we use the same memberships in B_{chem}, but this illustrative example serves to illustrate the importance of the known communities across networks. Due to the sparse structure of A_{gap}, the Leiden algorithm outputs multiple singleton communities which are used by CESSNA as known correspondences but *not* counted as seeds, shown as *Leiden Community with Singletons* in Figure 3. For a more fair comparison, we also merge all singleton communities into a single *super-community* along with other communities as the input of CESSNA; the results are shown as *Leiden Community without Singletons* in Figure 3. We compare both above with results of CESSNA using ground-truth communities (into motor-, sensory-, and interneurons), and results of the original SGM (Fishkind et al., 2019). CESSNA consistently outperforms SGM for any number of seeds using Leiden communities. Note that the Leiden algorithm is not deterministic, so different communities can be generated at every time. Testing across 100 Leiden community outputs reveals a maximum standard deviation of 0.05 in CESSNA's

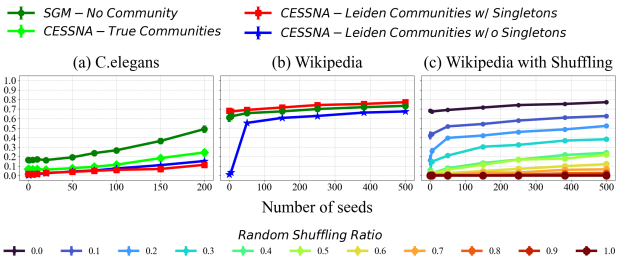


Figure 3: Graph Matching Ratios (y-axis) vs Number of Seeds (x-axis) of C.elegans and Wikipedia with Partial, Complete and Inferred Communities

matching accuracies, demonstrating its robust and effective utilization of inferred community structures.

Wikipedia Obtained from Wikipedia [2], the graph has $n = 1382$ vertices representing 1382 English articles from the 2-hop neighborhood of "Algebraic Geometry" along with the corresponding French documents. The articles are labeled into 6 disjoint classes that give rise to communities. Similar to the setting of (Fishkind et al., 2019), simulations are run on seed values $m \in \{0, 5, 50, 150, 250, 382, 500\}$, and they are averaged over 10 runs. Since the implementation of our algorithm assumes the m_i seeds are the first m_i elements of each group, we randomly shuffle the entire graph before the matching, which also helps with avoiding the identity bias introduced by the LAP solver (Jonker and Volgenant, 1988). To test our method's robustness to noisy information in community structures, we run experiments based on three sets of communities: true communities, communities inferred by Leiden clustering algorithm (Traag et al., 2019) on the graph of English documents (inferred then into the French graph) and *shuffled/errorful* communities. The shuffled communities are given by uniformly random shuffling a percentage of the original graph community labels. In addition, we perform shuffling for 10 independent runs for distinct shuffling ratio and number of seeds, and the standard deviations are presented above using error bars (accuracy $\pm$ 1 s.d.).

CESSNA's effectiveness with and without seeds As demonstrated in Figure 3(a), CESSNA achieves notable performance in seedless graph matching, attaining accuracy ratios of 0.1657 with Leiden communities in C.elegans graph. Remarkably, for the Wikipedia graph matching ratio shown in 3(b), when substituting true communities with partitions generated by the Leiden community detection algorithm (Traag et al., 2019), our method maintains a 0.61 matching accuracy without any seed initialization compared to the 0.68 from ground-truth communities. This also shows the flexibility of CESSNA to make use of the best avail-

[2]The original data is collected Dr. David J. Marchette in 2009 according to (Fishkind et al., 2019)

NUMBER OF SEEDS	0	5	50	150	250	382	500
CESSNA (true)	**0.68**	**0.67**	**0.69**	**0.72**	**0.74**	**0.75**	**0.77**
CESSNA (Leiden)	0.61	0.62	0.66	0.68	0.70	0.72	0.73
Grad-Align+	N/A	0.13	0.24	0.43	0.51	0.58	0.61

NUMBER OF SEEDS	0	1	5	10	20	50	75	100	150	200
CESSNA (true)	0.02	0.02	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.11
CESSNA (Leiden w/o singletons)	0.10	0.11	**0.12**	**0.12**	**0.11**	**0.13**	**0.17**	**0.18**	**0.21**	**0.24**
Grad-Align+	N/A	N/A	0.03	0.03	0.03	0.04	0.11	0.11	0.16	0.20

Table 1: Performance comparison with `Grad-Align+`: (top) Wikipedia; (bottom) C.elegans with different communities. Note accuracy does not include seed nodes.

able community information. As the number of seeds increase, our CESSNA consistently outperforms existing SGM algorithm (Fishkind et al., 2019) by a significant margin. Note that seeds can also make the algorithm more computationally efficient in Prop. 3.1.

CESSNA's robustness to community noise As demonstrated in Figure 3(c), CESSNA remains robust to noise in community information and continues to effectively utilize available seed information. With 10% noise and only 5 seeds, CESSNA attains a matching ratio of 0.4372, compared to 0.0346 by the SGM algorithm. Even with 30% noise, our CESSNA algorithm is still able to achieve an accuracy of 0.3826 with 500 nodes.

Comparison to state-of-the-art methods CESSNA is one of the first methods to directly apply community information while using seed information. Another method is *Grad-Align+* (Park et al., 2022), which feeds augmented node attributes—derived from centrality measures—alongside original attributes into a graph neural network to compute node similarities for alignment and has been shown to outperform other methods on many metrics. The comparison between CESSNA and *Grad-Align+* on real data is shown in 1 where CESSNA consistently outperforms *Grad-Align+* using either true communities or (the artificial) Leiden communities. Below we offer insights into the information provided by communities to better understand their purpose in network alignment.

4.3 Comparison to related methods

Since CESSNA jointly leverages both seed information and community structure, we focus our direct comparison on *Grad-Align+*, which also explicitly incorporates seeds together with label information. To more comprehensively illustrate CESSNA's performance, we additionally compare against related methods that use only seeds. To do this, we assume the knowledge of the communities and apply the matching algorithms within each community for a comparison with CESSNA (we note that this setup is not entirely balanced, as CESSNA continues to exploit information across communities, whereas methods restricted to within-community matching do not have access to such cross-community structure). Note that each method uses the seed set differently, and they

NUMBER OF SEEDS	1	5	10	20	50	75	100	150	200
CESSNA	**0.0417**	**0.0743**	**0.1363**	**0.3320**	**0.9987**	**1**	**1**	**1**	**1**
SeedGNN	0.0077	0.0213	0.0420	0.0817	0.2107	0.3380	0.4520	0.6637	0.8327
onehop6	0.0197	0.0287	0.0527	0.0833	0.1810	0.2757	0.3217	0.4807	0.6300
twohop3	0.0147	0.0193	0.0227	0.0443	0.1090	0.1943	0.2520	0.4523	0.6243
threehop2	0.0097	0.0070	0.0077	0.0120	0.0083	0.0100	0.0083	0.0110	0.0117
pgm	0.0100	0.0147	0.0133	0.0223	0.0357	0.0363	0.0377	0.0623	0.0763
sgm	0.0207	0.0407	0.0700	0.1153	0.3210	0.4917	0.6767	0.9087	0.9853
mgcn	0.0100	0.0200	0.0367	0.0667	0.1667	0.2533	0.3300	0.5067	0.6633

NUMBER OF SEEDS	1	5	10	20	50	75	100	150	200
CESSNA	**0.9960**	**1**	**1**	**1**	0.9952	**1**	**1**	**1**	**1**
SeedGNN	0.0100	0.0277	0.0627	0.2207	0.9973	**1**	**1**	**1**	**1**
onehop6	0.0223	0.0310	0.0540	0.1560	**1**	**1**	**1**	**1**	**1**
twohop3	0.0220	0.0277	0.0407	0.1093	0.9993	**1**	**1**	**1**	**1**
threehop2	0.0080	0.0127	0.0103	0.0113	0.0093	0.0130	0.0087	0.0080	0.0090
pgm	0.0100	0.0137	0.0160	0.0277	0.0597	0.0910	0.1453	0.2633	0.4527
sgm	0.0363	0.0693	0.2383	0.9623	**1**	**1**	**1**	**1**	**1**
mgcn	0.0100	0.0167	0.0333	0.0633	0.1633	0.2567	0.3367	0.5000	0.6600

Table 2: Top: Results on DCSBM graphs with correlation 0.3; Bottom: Results of DCSBM graphs with correlation 0.6. Note accuracy includes seed nodes.

do not necessarily force seeds to be matched correctly. The accuracies here are the overall accuracies including seed nodes, which is different from previous settings. We use the code directly from Yu et al. (2023) and compare with following baselines:

- *SeedGNN* (Yu et al. (2023)) learns from seeds how to propagate information from a small set of seed node pairs and iteratively infer accurate node correspondences between two graphs
- *D-hop* (Mossel and Xu (2019)) includes (1) *onehop6* that uses seeds as 1-hop witnesses, matching node pairs that share many consistent seeded neighbors, (2) *twohop3* that extends witness counting to 2-hop neighborhoods so more distant seeds can support matching, and (3) *threehop2* that further enlarges the witness region to 3 hops, letting faraway seeds contribute evidence.
- *PGM* (Kazemi et al. (2015)) starts from initial seed pairs and iteratively turns neighbors of matched pairs into new seeds.
- *MGCN* (Chen et al. (2020)) presents A multi-level graph convolutional network model that learns and aligns user representations to accurately predict anchor links connecting the same users on multiple networks.

Synthetic data For correlations $\rho = 0.3, 0.6$, we generate 10 pairs of correlated Degree-Corrected SBM (DCSBM) graphs with 3 communities and 100 nodes in each community. The probability parameters of this DCSBM are

$$\Lambda = \begin{bmatrix} 0.7 & 0.3 & 0.4 \\ 0.3 & 0.8 & 0.3 \\ 0.4 & 0.3 & 0.9 \end{bmatrix},$$

and the entries of degree-correction parameters θ are sampled uniformly from $(0.5, 1)$.

Real data We use the same real datasets introduced above — *C.elegans* and *Wikipedia* graphs. We can observe that across all experiments, our CESSNA performs well across all numbers of seeds. Even though

NUMBER OF SEEDS	0	5	50	150	250	382	500
CESSNA	**0.6800**	**0.6712**	**0.7012**	**0.7504**	**0.7870**	**0.8191**	**0.8532**
SeedGNN	0.0126	0.0109	0.0606	0.1861	0.2810	0.3956	0.4907
onehop6	0.0131	0.0125	0.0378	0.0950	0.1563	0.2127	0.2497
twohop3	0.0111	0.0098	0.0310	0.0587	0.0805	0.0965	0.1144
threehop2	0.0064	0.0077	0.0299	0.0624	0.0805	0.0888	0.0985
pgm	0.0043	0.0048	0.0121	0.0276	0.0429	0.0621	0.0763
sgm	0.0263	0.0310	0.0873	0.2380	0.3493	0.4625	0.5578
mgcn	0.0043	0.0050	0.0383	0.1114	0.1824	0.2779	0.3618

NUMBER OF SEEDS	1	5	10	20	50	75	100	150	200
CESSNA(true)	0.0235	0.0376	0.0455	0.0903	0.2038	0.2981	0.3905	0.5654	0.7367
CESSNA(Leiden)	**0.1032**	**0.1259**	**0.1515**	**0.1831**	**0.2695**	**0.3639**	**0.4675**	**0.6209**	**0.7763**
SeedGNN	0.0122	0.0301	0.0423	0.0796	0.1885	0.2867	0.3692	0.5602	0.7358
onehop6	0.0104	0.0133	0.0129	0.0158	0.0190	0.0269	0.0226	0.0273	0.0348
twohop3	0.0082	0.0147	0.0090	0.0122	0.0089	0.0147	0.0108	0.0161	0.0190
threehop2	0.0089	0.0068	0.0093	0.0093	0.0133	0.0133	0.0129	0.0169	0.0161
pgm	0.0107	0.0147	0.0169	0.0215	0.0247	0.0301	0.0351	0.0423	0.0427
sgm	0.0147	0.0312	0.0409	0.0810	0.1946	0.2857	0.3756	0.5631	0.7380
mgcn	0.0071	0.0215	0.0394	0.0681	0.1792	0.2652	0.3548	0.5341	0.7168

Table 3: Top: Results on Wikipedia graph; Bottom: Results on C.elegans. Note accuracy includes seed nodes.

using communities for C.elegans does not outperform (the artificial) inferred communities, it still performs well. We offer insights into this phenomenon in the next subsection.

4.4 Information Added by Communities

To understand the information across networks given by the communities, we employ the information-theoretic formulations capturing different aspects of overlap between graphs developed by (Felippe et al., 2024). Specifically, for given N-node graphs G_1 and G_2 with E_1 and E_2 edges respectively, divided into B communities, we consider following measures: For integers n, m, let $\left(\!\binom{n}{m}\!\right) = \binom{n+m-1}{m}$. Defining

$$\mathrm{I}_b^{\mathrm{meso}}(E_{12}=0) = \log \frac{\left(\!\binom{\binom{B}{2}+B}{E_1}\!\right)\left(\!\binom{\binom{B}{2}+B}{E_2}\!\right)}{\left(\!\binom{\binom{B}{2}+B}{E_1+E_2}\!\right)}$$

$$\mathrm{H}_{12,b} = \log\left(\!\binom{\binom{B}{2}+B}{E_1+E_2-E_{12,b}}\!\right), \mathrm{H}_{i,b} = \log\left(\!\binom{\binom{B}{2}+B}{E_i}\!\right),$$

$$E_{12,b} = \sum_{a \le b} \min(m_{1,a,b}, m_{2,a,b})$$

where $m_{i,a,b}$ counts the edges between communities a and b in G_i. Felippe et al. (2024) defines

$$\mathrm{MesoNMI}_b(G_1,G_2) = \frac{I_b^{\mathrm{meso}}(G_1,G_2) - I_b^{\mathrm{meso}}(E_{12}=0)}{(H_{1,b}+H_{2,b})/2 - I_b^{\mathrm{meso}}(E_{12}=0)}$$

where $I_b^{\mathrm{meso}}(G_1,G_2) = H_{1,b} + H_{2,b} - H_{12,b}$ is the raw measure that captures structural similarity exists between two graphs under partition b giving B communities, quantifying higher-level clustering structure while disregarding fine details like individual edge overlap. Intuitively, we would expect that more informative communities would lead to better matching performance. However, (at least empirically) this is not the case. MesoNMI is measuring how informative communities are to the graph structure at multiple scales. We see then that less informative communities often provide additional exogenous information to the CESSNA algorithm, and this information can be combined with the existing edge signal to provide better matchings. To explore this further, in Figure 4 we

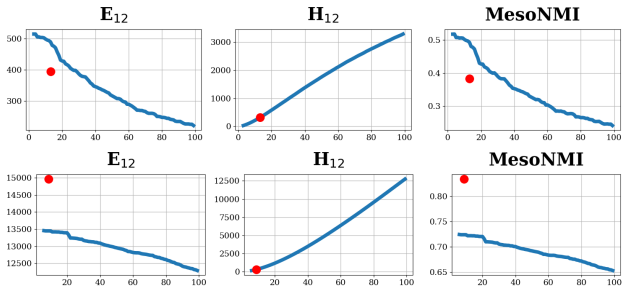


Figure 4: Top (resp., bottom): All metrics of C.elegans (resp., Wikipedia) graph with respect to the number of communities; Red dots illustrate the average Leiden communities results

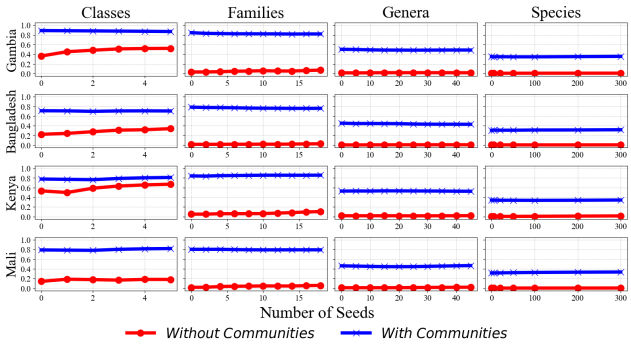


Figure 5: Graph Matching Ratios for Case and Control within Each Country by Rank (Phyla and super-kingdoms omitted)

present the changes in information with respect to the number of communities starting the true communities for both *C.elegans* and *Wikipedia* graphs (they have 3 and 6 true communities resp.).

Measures for a greater number of communities are constructed by dividing the largest community randomly into halves iteratively. We also calculate for Leiden communities without singletons averaged over 20 runs, shown in the red dots. Combined with Table 1, we can see this information-matching relation in action: less informative communities provide for better matching. This insight is practically useful as it allows practitioners to choose between multiple existing community structures to maximize CESSNA accuracy (choose the least informative communities, within reason).

5 MATCHING HIERARCHICAL MICROBIOMES

Hierarchical structures are common in network data, and we use a microbiome bacterial co-occurrence network to demonstrate the effectiveness of CESSNA. Mainstream microbiome network methods often rely on assumptions such as correlations (Friedman and Alm, 2012; Zhang et al., 2024) or Gaussian graphical

models (Ma, 2021), which face challenges from noise, limited data, and sparsity-driven inefficiencies. Our approach avoids these assumptions, enabling efficient inference across hierarchical levels without large sample sizes. We illustrate this with the following dataset.

Data The *msd16s* (Paulson et al., 2025) dataset contains *16S rRNA* gene sequencing data from 992 samples collected from healthy children and those with diarrhea across four low-income countries. Features include subject-level information (such as country) and the hierarchical taxonomy for 26043 operational taxonomic units (OTUs). The goal is to compare the gut microbiota of healthy and diseased (i.e., with diarrhea) individuals (Paulson et al., 2015).

Hierarchical Community Structure The biological taxonomy with hierarchical levels naturally providing ground-truth and informative community structures across classification levels: The given 26043 OTUs are classified into 754 species belonging to 164 genera that can be further grouped into 71 families, 18 classes, 9 phyla, and ultimately 2 super-kingdoms. For each rank, we compute co-occurrence matrices by taking the outer product of the matrices for all samples. Due to missing data, each of these categories includes an *N/A* class (treated as a separate label class), comprising 22.76% of the entire dataset. A small number of inconsistently classified samples are omitted in our analysis as they do not substantially impact our results owing to the robustness of our algorithm. Note that there are 1, 6, 7, 41, 76 singleton communities in the clusters of phyla, classes, families, genera and species respectively; these will inflate the accuracy of `CESSNA` versus matching without communities (though the matching accuracy is significantly higher for `CESSNA` even accounting for this). We include them here as they are the true cluster labels, and are present in all comparisons in Figure 6; the entire figure is attached in appendix.

Within and Cross-country Inferences Figure 5 presents matching results for co-occurrence matrices of the control and case groups for each country on selected ranks. The entire figure is in the appendix. The node correspondences identify the presence of same OTU groups at each rank across sample groups. Compared to the faint signal marked by the red lines, the high graph match ratios with community information indicate that microbial family co-occurrence structures are largely preserved between healthy and diarrheal states within a country. In other words, diarrhea does not disrupt the overall co-occurrence patterns among children from the same country. Figure 6 presents matching ratios for country pairs.

Although geography influences bacterial interaction

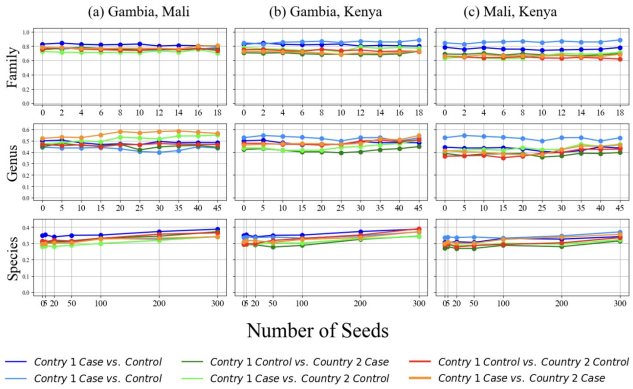


Figure 6: Graph Matching Ratios Within and Across Countries for Control and Case Groups (selected countries)

networks (Wang et al., 2022), our results show that co-occurrence structures remain relatively robust to geography and local environment. However, the lower Case vs. Control accuracy across countries compared to within countries suggests geography still affects structural similarity. For some country pairs (e.g., Mali and Kenya), across-country accuracy is consistently lower, highlighting geography's role in microbial co-occurrence structures, especially at the family and genus levels. Yet this effect is not universal, as seen with Mali and Gambia, where matching is similar within and across countries, suggesting geography is seemingly not the main driver. Within-country similarity and between-country divergence across taxonomic levels suggest that microbial co-occurrence patterns are robust to diarrhea status within a country, while cross-country comparisons show mixed sensitivity to geographic and environmental context.

6 CONCLUSION

We present a novel algorithm `CESSNA` for the network alignment (i.e., graph matching) problem, and demonstrate its innovative applicability to graph matching tasks for multiscale data. Experimental results on both synthetic and real-world datasets show that this approach consistently outperforms existing methods. In addition, our algorithm exhibits adaptability to diverse conditions: it seamlessly integrates seed information, and leverages the information contained in noisy ground-truth communities when available. In addition to traditional applications of vertex de-anonymization, the proposed algorithm shows its effective usage in inference tasks for biological datasets unexplored by graph matching. Indeed, the simplicity and adaptability of our method make it a powerful tool for hierarchical data analysis, with strong potential for broader applications in network analysis.

References

Aflalo, Y., Bronstein, A., and Kimmel, R. (2015). On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences*, 112(10):2942–2947.

Babai, L. (2016). Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 684–697, New York, NY, USA. Association for Computing Machinery.

Bai, X., Yu, H., and Hancock, E. (2004). Graph matching using spectral embedding and alignment. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 398–401 Vol.3.

Bartomeus, I., Gravel, D., Tylianakis, J. M., Aizen, M. A., Dickie, I. A., and Bernard-Verdier, M. (2016). A common framework for identifying linkage rules across different types of interactions. *Functional Ecology*, 30(12):1894–1903.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Chen, H., YIN, H., Sun, X., Chen, T., Gabrys, B., and Musial, K. (2020). Multi-level graph convolutional networks for cross-platform anchor link prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1503–1511, New York, NY, USA. Association for Computing Machinery.

Chen, L., Vogelstein, J. T., Lyzinski, V., and Priebe, C. E. (2015). A joint graph inference case study: the c.elegans chemical and electrical connectomes.

Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004). Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298.

Ding, Y. and Wolkowicz, H. (2009). A low-dimensional semidefinite relaxation for the quadratic assignment problem. *Mathematics of Operations Research*, 34(4):1008–1022.

Escolano, F., Hancock, E., and Lozano, M. (2011). Graph matching through entropic manifold alignment. In *CVPR 2011*, pages 2417–2424.

Fang, F., Sussman, D. L., and Lyzinski, V. (2018). Tractable graph matching via soft seeding. *arXiv preprint arXiv:1807.09299*.

Felippe, H., Battiston, F., and Kirkley, A. (2024). Network mutual information measures for graph similarity. *Communications Physics*, 7(1):335.

Fishkind, D. E., Adali, S., Patsolic, H. G., Meng, L., Singh, D., Lyzinski, V., and Priebe, C. E. (2019). Seeded graph matching. *Pattern recognition*, 87:203–215.

Fortunato, S. and Castellano, C. (2007). Community structure in graphs. *arXiv preprint arXiv:0712.2716*.

Frank, M., Wolfe, P., et al. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.

Friedman, J. and Alm, E. J. (2012). Inferring correlation networks from genomic survey data.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137.

Jonker, R. and Volgenant, T. (1988). A shortest augmenting path algorithm for dense and sparse linear assignment problems. In *DGOR/NSOR: Papers of the 16th Annual Meeting of DGOR in Cooperation with NSOR/Vorträge der 16. Jahrestagung der DGOR zusammen mit der NSOR*, pages 622–622. Springer.

Kazemi, E., Hassani, S. H., and Grossglauser, M. (2015). Growing a graph matching from a handful of seeds. *Proceedings of the VLDB Endowment*, 8(10):1010–1021.

Lall, S., Grün, D., Krek, A., Chen, K., Wang, Y.-L., Dewey, C. N., Sood, P., Colombo, T., Bray, N., MacMenamin, P., et al. (2006). A genome-wide map of conserved microrna targets in c. elegans. *Current biology*, 16(5):460–471.

Lancichinetti, A., Radicchi, F., Ramasco, J. J., and Fortunato, S. (2011). Finding statistically significant communities in networks. *PloS one*, 6(4):e18961.

Lee, J., Cho, M., and Lee, K. M. (2010). A graph matching algorithm using data-driven markov chain monte carlo sampling. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, page 2816–2819, USA. IEEE Computer Society.

Legras, G., Loiseau, N., Gaertner, J.-C., Poggiale, J.-C., Ienco, D., Mazouni, N., and Mérigot, B. (2019). Assessment of congruence between co-occurrence and functional networks: A new framework for revealing community assembly rules. *Scientific Reports*, 9(1):19996.

Li, Y., Dong, M., and Hua, J. (2007). A gaussian mixture model to detect clusters embedded in feature subspace.

Lim, S. H., Chen, Y., and Xu, H. (2014). Clustering from labels and time-varying graphs. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, page 1188–1196, Cambridge, MA, USA. MIT Press.

Lin, L., Liu, X., and Zhu, S.-C. (2010). Layered graph matching with composite cluster sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1426–1442.

Liu, J., Wright, S., Ré, C., Bittorf, V., and Sridhar, S. (2014). An asynchronous parallel stochastic coordinate descent algorithm. In *International Conference on Machine Learning*, pages 469–477. PMLR.

Lyzinski, V., Fishkind, D. E., Fiori, M., Vogelstein, J. T., Priebe, C. E., and Sapiro, G. (2016). Graph matching: Relax at your own risk. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1):60–73.

Lyzinski, V., Fishkind, D. E., and Priebe, C. E. (2014). Seeded graph matching for correlated erdös-rényi graphs. *J. Mach. Learn. Res.*, 15(1):3513–3540.

Ma, J. (2021). Joint microbial and metabolomic network estimation with the censored gaussian graphical model. *Statistics in biosciences*, 13(2):351–372.

Maron, H. and Lipman, Y. (2018). (probably) concave graph matching. *Advances in Neural Information Processing Systems*, 31.

Mazein, I., Rougny, A., Mazein, A., Henkel, R., Gütebier, L., Michaelis, L., Ostaszewski, M., Schneider, R., Satagopam, V., Jensen, L. J., et al. (2024). Graph databases in systems biology: a systematic review. *Briefings in Bioinformatics*, 25(6):bbae561.

Mossel, E. and Xu, J. (2019). Seeded graph matching via large neighborhood statistics. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, page 1005–1014, USA. Society for Industrial and Applied Mathematics.

Mossel, E. and Xu, J. (2020). Seeded graph matching via large neighborhood statistics. *Random Structures & Algorithms*, 57(3):570–611.

Narayanan, A. and Shmatikov, V. (2009). Deanonymizing social networks. In *2009 30th IEEE Symposium on Security and Privacy*, pages 173–187.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.

Park, J.-D., Tran, C., Shin, W.-Y., and Cao, X. (2022). Gradalign+: Empowering gradual network alignment using attribute augmentation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 4374–4378, New York, NY, USA. Association for Computing Machinery.

Patsolic, H. G. (2020). *GRAPH MATCHING AND VERTEX NOMINATION*. PhD thesis, Johns Hopkins University.

Paulson, J. N., Bravo, H. C., and Pop, M. (2025). *msd16s: Healthy and moderate to severe diarrhea 16S expression data*. Bioconductor. Bioconductor package version 1.26.0.

Paulson, J. N., Bravo, H. C., Pop, M., and biocViews ExperimentData, S. (2015). Package 'msd16s'.

Richtárik, P. and Takáč, M. (2016). Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156:433–484.

Sah, P., Singh, L. O., Clauset, A., and Bansal, S. (2014). Exploring community structure in biological networks with random graphs. *BMC bioinformatics*, 15:1–14.

Sahni, S. and Gonzalez, T. (1976). P-complete approximation problems. *J. ACM*, 23(3):555–565.

Singh, R., Xu, J., and Berger, B. (2008). Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 105(35):12763–12768.

Tokala, S., Enduri, M. K., Lakshmi, T. J., and Hajarathaiah, K. (2025). Evaluating Community Detection Algorithms: A Focus on Effectiveness and Efficiency. *Journal of Scientometric Research*, 14(1):62–74.

Traag, V. A., Waltman, L., and Van Eck, N. J. (2019). From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

Valouev, A., Ichikawa, J., Tonthat, T., Stuart, J., Ranade, S., Peckham, H., Zeng, K., Malek, J. A., Costa, G., McKernan, K., et al. (2008). A high-resolution, nucleosome position map of c. elegans reveals a lack of universal sequence-dictated positioning. *Genome research*, 18(7):1051–1063.

Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H., and Chklovskii, D. B. (2011). Structural properties of the caenorhabditis elegans neuronal network. *PLoS computational biology*, 7(2):e1001066.

Vogelstein, J. T., Conroy, J. M., Lyzinski, V., Podrazik, L. J., Kratzer, S. G., Harley, E. T., Fishkind, D. E., Vogelstein, R. J., and Priebe, C. E. (2015). Fast approximate quadratic programming for graph matching. *PLOS ONE*, 10(4):1–17.

Wang, Z., Zhang, C., Li, G., and Yi, X. (2022). The influence of species identity and geographic locations on gut microbiota of small rodents. *Frontiers in Microbiology*, 13:983660.

Yartseva, L. and Grossglauser, M. (2013a). On the performance of percolation graph matching. In *Proceedings of the First ACM Conference on Online Social Networks*, COSN '13, page 119–130, New York, NY, USA. Association for Computing Machinery.

Yartseva, L. and Grossglauser, M. (2013b). On the performance of percolation graph matching. In *Proceedings of the first ACM conference on Online social networks*, pages 119–130.

Yu, L., Xu, J., and Lin, X. (2021). Graph matching with partially-correct seeds. *Journal of Machine Learning Research*, 22(280):1–54.

Yu, L., Xu, J., and Lin, X. (2023). Seedgnn: graph neural network for supervised seeded graph matching. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Zaslavskiy, M., Bach, F., and Vert, J.-P. (2009). A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2227–2242.

Zhang, S., Fang, H., and Hu, T. (2024). fastcclasso: a fast and efficient algorithm for estimating correlation matrix from compositional data. *Bioinformatics*, 40(5):btae314.

Zhou, X., Liang, X., Du, X., and Zhao, J. (2017). Structure based user identification across social networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(6):1178–1191.

Zhu, Y., Qin, L., Yu, J. X., Ke, Y., and Lin, X. (2011). High efficiency and quality: large graphs matching. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1755–1764.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Community-Enhanced Semi-seeded Network Alignment (CESSNA): A Robust Method with Application to Microbiome Networks: Supplementary Materials

1 Proof of Theorem 1

Definition 1 (Correlated Stochastic Blockmodel (CorrSBM))**.** Suppose the vertex set $V = \{1, 2, \ldots, n\}$ is partitioned into K blocks $C_1, \ldots, C_K$, with a symmetric block probability matrix $\mathbf{P} \in [0,1]^{K \times K}$ and a symmetric blockwise correlation matrix $\mathbf{R} \in [0,1]^{K \times K}$. Let $\mathbf{G}_1, \mathbf{G}_2$ be two random graphs with adjacency matrices A and B respectively. Then $(A, B) \sim \text{CorrSBM}(\mathbf{P}, \mathbf{R}, g)$ if:

1. For all pairs $1 \leq u < v \leq n$, let $g(u), g(v) \in \{1, \ldots, K\}$ denote the block memberships of u and v.
2. $A[u, v] \sim \text{Bernoulli}(P[g(u), g(v)])$ are independent for all $u < v$.
3. $B[u, v] \sim \text{Bernoulli}(P[g(u), g(v)])$ are independent for all $u < v$.
4. For each $1 \leq u < v \leq n$, $\text{corr}(A[u, v], B[u, v]) = R[g(u), g(v)]$.
5. For $(u, v) \neq (k, l)$, the pairs $(A[u, v], B[u, v])$ and $(A[k, l], B[k, l])$ are independent.

Define $Q[u, v] = R[g(u), g(v)]P[g(u), g(v)](1 - P[g(u), g(v)])$ as the covariance for the (u, v) edge. Then each block of CorrSBM falls under CorrER, and the overall CorrSBM is a generalization of CorrER.

Note that for ease of exposition below, below we will consider the setting where all communities are the same order $c = |\{v \in V \text{ s.t. } g(v) = i\}|$. Let $K = n/c$. In all results below, we will be making use of the two quantities (for $(A, B) \sim \text{CorrSBM}(\mathbf{P}, \mathbf{R}, g)$),

$$C = \min_{a,b \in [K]} R[a, b]P[a, b](1 - P[a, b]), \quad \varepsilon = \max_{a,b \in [K]} \underbrace{[3P[a, b](1 - P[a, b]) + 2R[a, b]]}_{:= \varepsilon_{a,b}}.$$

Theorem 1.0.1. Suppose $(A, B) \sim \text{CorrSBM}(\mathbf{P}, \mathbf{R}, g)$. Let $\delta \in (0, 1/2)$. There exist constants $a > 0$ and $b > 0$ such that if $\ell = \sqrt{aKc^{1+2\delta}}$ and $m = bc^{1-\delta} \log c / \sqrt{K}$, and if for all $i, j \in [K]$, Suppose that for all $i, j \in [K]$,

$$\mathbb{E}[\text{trace}(A_{i,j} D B_{i,j}(Q - P))] \geq C \, \text{trace}(D)(\text{trace}(Q) - \text{trace}(P))$$

for all permutation matrices D, P, Q with $\text{trace}(P) \leq c - m$, $\text{trace}(D) \geq \ell$, and $\text{trace}(Q) \geq c - m$. If $m = o(c/(K \log^2 c)), m = \omega(\sqrt{c \log c})$, then with probability at least

$$1 - \exp\left\{-\Theta(c \log^2 c)\right\} - \exp\left\{-\Theta(c^{1+\delta} \log c)\right\} - \exp\left\{-\Theta(c)\right\}$$

CESSNA will converge to the true correspondence, I, in at most two steps starting from any $P = \oplus_i P_i$ with all $\ell \leq \text{trace}(P_i) \leq c - 2m$.

Theorem 1 follows from Corollary 1, Proposition 1, and Proposition 2 below. Our Theorem 1.0.1 is an analogue of Theorem 3 of [Fang et al., 2018], and we closely follow below the proof of Theorem 3 in [Fang et al., 2018], adapted here to our present SBM and CESSNA setting.

1.1 One Step Convergence

We first establish one step convergence results, and some key preliminary results. Let $A_{i,j}$ (resp., $B_{i,j}$) denote the subgraph of A (resp., B) connecting communities i and j (with $i = j$ allowed).

Lemma 1.1.1 (Variant of Lemma 8 in [Fang et al., 2018])**.** Let $D \in \mathcal{D}_c$ (the set of $c \times c$ doubly stochastic matrices), and $P, Q \in \Pi_c$ (the set of $c \times c$ permutation matrices) be such that $\text{trace}(D) = m \leq c$, and

$$c - k = \text{trace}(P) \leq \text{trace}(Q) = c - \ell.$$

Let $d = \|P - Q\|_1$. For $(A, B) \sim \text{CorrSBM}(\mathbf{P}, \mathbf{R}, g)$, suppose that for all $i, j \in [K]$,

$$\mathbb{E}[\text{trace}(A_{i,j} D B_{i,j}(Q - P))] > C(k - l)m.$$

It then holds that for all $i, j \in [K]$,

$$\mathbb{P}[\exists\, i, j \in [K], \text{ s.t. } \text{trace}(A_{i,j} D B_{i,j}(Q - P)) \leq 0] \leq 2K^2 \exp\left\{-\frac{C^2(k-\ell)^2 m^2}{18432\varepsilon c d}\right\}.$$

For the case of $Q = I$, we have that

$$\mathbb{P}[\exists\, i, j \in [K], \text{ s.t. } \text{trace}(A_{i,j} D B_{i,j}(I - P)) < 0] \leq 2K^2 \exp\left\{-\frac{C^2 k m^2}{6144\varepsilon c}\right\}.$$

Proof. Note that for each $\{u, v\}$, we can write the Bi-Bernoulli pair $(A[u, v], B[u, v])$ as a function of three independent Bernoulli random variables:

$$(A[u, v], B[u, v]) \stackrel{d}{=} (Z_0[u, v],\ (1 - Z_0[u, v])Z_1[u, v] + Z_0[u, v]Z_2[u, v])$$

where

$$Z_0[u, v] \sim \text{Bern}\left(P[g(u), g(v)]\right), \quad Z_1[u, v] \sim \text{Bern}\left(P[g(u), g(v)](1 - R[g(u), g(v)])\right),$$

and

$$Z_2[u, v] \sim \text{Bern}\left(P[g(i), g(j)] + R[g(i), g(j)](1 - P[g(i), g(j)])\right).$$

Note that $\text{Var}(Z_0[u, v] + Z_1[u, v] + Z_2[u, v]) \leq \varepsilon_{g(u),g(v)} \leq \varepsilon$ as defined above, and that $\text{Var}(A[u, v]) = \text{Var}(B[u, v]) \geq C$. We will next use Proposition 6 in [Fang et al., 2018] (adapted there from [Alon et al., 1997]) to provide a sharp bound on the random variable for a given $\{i, j\}$ pair (where $i = j$ is allowed)

$$X = \text{trace}(A_{i,j} D B_{i,j}(Q - P)) = \sum_{u=1}^{c}\sum_{w=1}^{c}\sum_{v=1}^{c} A_{i,j}[u, w] D[w, v](B_{i,j}[v, \tau(u)] - B_{i,j}[v, \sigma(u)])$$

where τ is the permutation associated with Q and σ the permutation associated with P. This X is a function of independent Bernoulli random variables with different parameters based on the block assignments, and X depends on the entries of $A_{i,j}$ and $B_{i,j}$ with indices in the set (when $i \neq j$, these are ordered pairs)

$$\left\{\{u, w\}, \{\tau(u), v\}, \{\sigma(u), v\} : u, v, w \in [c],\ \tau(u) \neq \sigma(u)\right\}$$

which has cardinality upper bounded by $6cd$ (double counting to account for directedness when $i \neq j$); and X is a function of at most $18cd$ independent Bernoulli random variables (the associated Z's). Note that we can say that X is a function of precisely $18cd$ independent Bernoulli random variables by considering X also as a function of additional, independent $Z'_k[u, v]$ triplets (identically distributed to one of the $Z_k[u, v]$ triplets) where X's value is invariant to changes in the $Z'_k[u, v]$. In the case that $\tau(i) = i$ for all i, this set has cardinality bounded by $2cd = 2ck$, as $\{\{u, v\} : \sigma(u) \neq u\} = \{\{\sigma(u), v\} : \sigma(u) \neq u\}$.

To get a concentration bound for X using Proposition 6 in [Fang et al., 2018], we need to have a bound on the sum of the variances of all the independent Bernoulli random variables that X depends on. Note that changing any one of the $Z_k[u, v]$ random variables can change X by at most $M = 4$ (it can change both the associated $A[u, v]$ and $B[u, v]$, and changing either of these can change X by at most 2). Letting $\gamma^2 =$

$M^2(\sum \mathrm{Var}(Z_i[u,v]) + \sum \mathrm{Var}(Z_i'[u,v]))$ (where the sums are over all $6cd$ Bernoulli's that X depends on), and noting that $C \leq \mathrm{Var}(Z_0[u,v])$ for all $\{u,v\}$, we have that

$$M^2 cdC \leq \gamma^2 \leq 18M^2 cd \max_{a,b\in[K]} \varepsilon_{ab} \leq 288cd\varepsilon =: \gamma'^2,$$

where $\varepsilon_{ab} = 3P[a,b](1-P[a,b]) + 2R[a,b]$ for block-pair (a,b).

Applying Proposition 6 of [Fang et al., 2018] yields that for $0 < t < 2\gamma/M$, we have

$$\mathbb{P}\left[|X - \mathbb{E}X| \geq t\gamma\right] \leq 2e^{-t^2/4}.$$

We will apply this here with $t = \frac{C(k-l)m}{4\gamma'}$. First we must verify that this t is less that $2\gamma/M$. To this end, note that

$$C(k-l)m/4\gamma' < \frac{2\gamma}{M} \text{ is equivalent to } m < \frac{2\gamma\gamma'}{C(k-\ell)},$$

and since $C \leq \epsilon$ and $d \geq k-\ell$, we have

$$m \leq c < \frac{2cCd}{C(k-\ell)} \leq \frac{2\gamma^2}{(k-\ell)C} \leq \frac{2\gamma\gamma'}{(k-\ell)C},$$

as desired. Using this t, we have that

$$\begin{aligned}\mathbb{P}[X \leq 0] &\leq \mathbb{P}\left[|X - \mathbb{E}X| \geq \frac{\mathbb{E}X}{\gamma}\gamma\right] \leq \mathbb{P}\left[|X - \mathbb{E}X| \geq \frac{C(k-l)m}{4\gamma'}\gamma\right] \\ &\leq 2\exp\left(-\frac{C^2(k-l)^2m^2}{16 * \gamma'^2 * 4}\right) = 2\exp\left(-\frac{C^2(k-l)^2m^2}{18432cd\varepsilon}\right),\end{aligned}$$

and a union bound over $\{i,j\}$ finishes the proof. □

We next provide a variant of Theorem 9 of [Fang et al., 2018], the one step convergence result.

Theorem 1.1.1 (Analogue of Theorem 9 of [Fang et al., 2018])**.** For each $i \in [K]$, let $P_i \in \Pi_c$ have $\mathrm{trace}(P_i) \geq m$. Let $(A,B) \sim \mathrm{CorrSBM}(\mathbf{P},\mathbf{R},g)$ be such that for all $i,j \in [K]$ and all $P \in \Pi_c$,

$$\mathbb{E}[\mathrm{trace}(A_{i,j}P_iB_{i,j}(I-P))] \geq Cm\,\mathrm{trace}(I-P)$$

If

$$m \geq \sqrt{\frac{12288\varepsilon c\log c}{C^2}},$$

then with probability at least

$$1 - 6K^3\exp\left\{-\left(\frac{C^2m^2}{12288\varepsilon c}\right)\right\},$$

the first step of `CESSNA` in the j-th search direction, starting at $\oplus_i P_i$, yields the identity permutation matrix as the solution.

Proof. We consider here `CESSNA` optimizing over the j-th block of the global permutation. Consider starting position given by $P = \oplus_i P_i$. By Lemma 1.1.1 above, for each permutation matrix $P \in \Pi_c$ with $\mathrm{trace}(P) = c-k$, we have that (considering $D = P_i$ in Lemma 1.1.1 and considering an additional union over these D to account for all P_i which provides the extra factor of K),

$$\mathbb{P}\Big[\exists\, i,j \in [K] \text{ s.t. } \mathrm{trace}(A_{i,j}P_iB_{i,j}(I-P)) < 0\Big] \leq 2K^3\exp\left\{-\frac{C^2km^2}{6144\varepsilon c}\right\}.$$

The number of such $P \in \Pi_c$ for each k is at most c^k (let the set of such P be denoted $\Pi_{c,k}$). Applying a union bound over all $k \geq 2$, we have that

$$\begin{aligned}
&\mathbb{P}\left[\exists\, i,j \in [K] \text{ s.t.} \min_{P \neq I} \operatorname{trace}(A_{i,j} P_i B_{i,j}(I-P)) < 0\right] \\
&\quad = \mathbb{P}\left[\exists\, i,j \in [K] \text{ s.t. } \exists\, k > 0, \exists\, P \in \Pi_{c,k} \text{ with } \operatorname{trace}(A_{i,j} P_i B_{i,j}(I-P)) < 0\right] \\
&\quad \leq \sum_{k=2}^{c} c^k 2K^3 \exp\left\{-\frac{C^2 k m^2}{6144\varepsilon c}\right\} \\
&\quad \leq \sum_{k=2}^{c} 2K^3 \exp\left\{k\left(\log c - \frac{C^2 m^2}{6144\varepsilon c}\right)\right\} \\
&\quad \leq \sum_{k=2}^{c} 2K^3 \exp\left\{-k\left(\frac{C^2 m^2}{12288\varepsilon c}\right)\right\} \\
&\quad \leq 2K^3 \exp\left\{\log c - 2\left(\frac{C^2 m^2}{12288\varepsilon c}\right)\right\} \\
&\quad \leq 2K^3 \exp\left\{-\left(\frac{C^2 m^2}{12288\varepsilon c}\right)\right\},
\end{aligned}$$

and with high probability, the line-search step of CESSNA will choose the identity matrix for its search direction.

To understand the step size, we consider (using the notation from Section 3.1) the size of the step taken in the direction $Q_j = I$

$$\beta_j = \min_{\beta \in [0,1]} \operatorname{tr}\left(A P_{j,\beta} B^T (P_{j,\beta})^T\right); \tag{1}$$

To show the optimum is achieved at $\beta_j = 0$, we take the derivative of $\operatorname{tr}\left(A P_{j,\beta} B^T (P_{j,\beta})^T\right)$ with respect to β and show that it is negative at 0 and at 1 (and hence $\operatorname{tr}\left(A P_{j,\beta} B^T (P_{j,\beta})^T\right)$ is decreasing in β). To this end, we compute

$$\begin{aligned}
\frac{d}{d\beta} \operatorname{tr}\left(A P_{j,\beta} B^T (P_{j,\beta})^T\right)\Big|_0 =& 2\sum_{i \neq j} \operatorname{tr}(A_{i,j} P_i B_{i,j}(P_j^\top - I)) \\
&- 2\operatorname{tr}(A_{jj} B_{jj}) + \operatorname{tr}(A_{jj} P_j B_{jj}) + \operatorname{tr}(A_{jj} B_{jj} P_j^T)
\end{aligned}$$

The first line of the above is negative with high probability by what was shown above; the second line is also negative with high probability by considering $D = I$ in Lemma 1.1.1. Now,

$$\begin{aligned}
\frac{d}{d\beta} \operatorname{tr}\left(A P_{j,\beta} B^T (P_{j,\beta})^T\right)\Big|_1 =& 2\sum_{i \neq j} \operatorname{tr}(A_{i,j} P_i B_{i,j}(P_j^\top - I)) \\
&- \operatorname{tr}(A_{jj} P_j B_{jj}) - \operatorname{tr}(A_{jj} B_{jj} P_j^T) + 2\operatorname{tr}(A_{jj} P_j B_{jj} P_j^T)
\end{aligned}$$

Again, the first line of the above is negative with high probability by what was shown above; the second line is also negative with high probability by considering $D = P_j$ in Lemma 1.1.1. A union over all choices for D ($D = P_i$ for all $i \neq j$ and $D = I$ and $D = P_j$) would then yield the probability that the step size is not 1 is bounded above by $4K^3 \exp\left\{-\left(\frac{C^2 m^2}{12288\varepsilon c}\right)\right\}$.

Combining the above, we have that with high probability, the search direction will be I and CESSNA will take a full step to I as desired. □

We next prove the analogue of Corollary 10 from [Fang et al., 2018], our one step convergence result.

Corollary 1.1.1 (analogue of Corollary 10 from [Fang et al., 2018])**.** Let $(A, B) \sim \text{CorrSBM}(\mathbf{P}, \mathbf{R}, g)$ as in Definition 1, and let

$$\ell = \frac{cC^2}{K\varepsilon \log^2 c} = \omega(\sqrt{c \log c}).$$

Suppose that for all $i, j \in [K]$ and all permutation matrices D^0, P with $\text{trace}(D^0) = m \geq c - \ell$,

$$\mathbb{E}\big[\,\text{trace}(A_{i,j} D^0 B_{i,j}(I - P))\big] > C\ \text{trace}(D^0)\,\text{trace}(I - P).$$

Then, with probability at least

$$1 - 6K^4 \exp\left\{-\Theta\left(\frac{cC^2}{\epsilon}\right)\right\},$$

for all permutation matrices $\oplus_j D_j^0$ such that $\text{trace}(D_j^0) \geq c - \ell$ for all j, the first step of the `CESSNA` procedure in any component direction started at $\oplus_j D_j^0$ will yield the identity matrix.

Proof. Apply Theorem 1.1.1 above. We will consider a union bound over all permutation matrices $D = \oplus D_j$ where for each j, there is less than $c - \text{trace}(D_j) \leq \ell = \frac{cC^2}{K\varepsilon \log^2 c}$ non-fixed points. First note that the number of such D is at most $c^{2\ell K}$. We then obtain the stated probability bound:

$$\mathbb{P}(\text{the first step in any direction started at any such } D = \oplus_j D_j \text{ does not yield } I)$$
$$\sum_{\text{direction } = i} 6K^3 \exp\left\{2\ell K \log c - \frac{C^2(c-\ell)^2}{12288\varepsilon c}\right\} = 6K^4 \exp\left\{-\Theta\left(\frac{cC^2}{\epsilon}\right)\right\}$$

□

1.2 Two Step Convergence

To prove the two step convergence, we next turn our attention to a variant of Proposition 11 of [Fang et al., 2018].

Proposition 1.2.1 (variant of Proposition 11 of [Fang et al., 2018]; Step Direction for CorrSBM)**.** Suppose $(A, B) \sim \text{CorrSBM}(\mathbf{P}, \mathbf{R})$ as in Definition 1, and let $\delta \in (0, 1/2)$ be fixed. Set

$$\ell^2 = 18432 \frac{K c^{1+2\delta} \epsilon^2}{C^4} \qquad \text{and} \qquad m = \frac{C^2 c^{1-\delta} \log(c)}{\sqrt{K}\epsilon}.$$

Suppose that for all $i, j \in [K]$,

$$\mathbb{E}\big[\,\text{trace}(A_{i,j} D B_{i,j}(I - Q))\big] \geq C\ \text{trace}(D)(c - \text{trace}(Q))$$

for all permutation matrices D, Q with $\text{trace}(D) \geq \ell$ and $\text{trace}(Q) \leq c - m$. Then, with probability at least

$$1 - 2K^4 \exp\left\{-c^{1+\delta} \log c\right\}$$

starting at any permutation matrix $\oplus D_j$ (with $\text{trace}(D_j) \geq \ell$), the first step of the `CESSNA` procedure in any of the K components will be in the direction of a permutation matrix D' with $\text{trace}(D') > c - m$.

Proof. We again closely follow the analogous proof in [Fang et al., 2018]. For integer $b \leq c$, define

$$\Pi_{c,\leq b} = \cup_{k \leq b} \Pi_{c,k},\ \Pi_{c,\geq b} = \cup_{k \geq b} \Pi_{c,k}$$

and consider the event

$$\mathcal{E} = \big\{\forall D' \in \Pi_{c,\leq m-1},\ \exists i, j \in [K],\ \exists D \in \Pi_{c,\leq n-\ell},\ \exists Q \in \Pi_{c,\geq m},\ \text{ s.t. } \text{trace}(A_{ij} D B_{ij}(D' - Q)^T) \leq 0\big\}$$

(so that $\text{trace}(D) \geq \ell$, and $\text{trace}(Q) \leq c - m$, and $\text{trace}(D') \geq c - m + 1$). Here $\mathcal{E}^c$ is the event that there exists a D' (good direction) such that for any i, j and any such D (our starting place) and Q (a potential bad search direction), D' is a better search direction than Q; this is because the search direction here minimizes (over P)

$$-\sum_{i \neq j} \text{tr}(A_{j,i} P_i^{(t)} B_{j,i}^T P^T) - \sum_{i \neq j} \text{tr}(A_{i,j}^T P_i^{(t)} B_{i,j} P^T) - \text{tr}(A_{j,j} P_j^{(t)} B_{j,j}^T P^T) - \text{tr}(A_{j,j}^T P_j^{(t)} B_{j,j} P^T),$$

and D' minimizes each term better than Q. Note that D' need not be the optimal search direction, but it does exclude all Q with $\text{trace}(Q) \leq c - m$ from the optimal direction.

In particular, if $\mathcal{E}$ holds, then it hold for $D' = I$ (which we use below). A union bound and application of Lemma 1.1.1 for CorrSBM yielding (upper bounding $d \leq \eta + \xi$ below)

$$\begin{aligned}
\mathbb{P}(\mathcal{E}) &\leq \sum_{\xi=2}^{c-\ell} \sum_{\eta=m}^{c} 2K^4 \exp\left\{(\eta+\xi)\log c - \frac{C^2(c-\xi)^2\eta}{6144\epsilon c}\right\} \quad \text{(setting } D' = I \text{ here)} \\
&\leq \sum_{\eta=m}^{c} 2K^4 \exp\left\{(\eta+c+1)\log c - \frac{C^2\ell^2\eta}{6144\epsilon c}\right\} \\
&\leq 2K^4 \exp\left\{2(c+1)\log c - \frac{C^2\ell^2 m}{6144\epsilon c}\right\} \\
&\leq 2K^4 \exp\left\{2(c+1)\log c - \frac{C^4\ell^2 c^{1-\delta}\log c}{6144\sqrt{K}\epsilon^2 c}\right\} \\
&= 2K^4 \exp\left\{2(c+1)\log c - 3\sqrt{K}c^{1+\delta}\log c\right\} \leq 2K^4 \exp\left\{-c^{1+\delta}\log c\right\}
\end{aligned}$$

as desired. □

We lastly turn our attention to the analogue of Proposition 12 in [Fang et al., 2018].

Proposition 1.2.2 (analogue of Proposition 12 in [Fang et al., 2018]; Step Size for CorrSBM)**.** Suppose $(A, B) \sim \text{CorrSBM}(\mathbf{P}, \mathbf{R})$ as in Definition 1. Let $\delta \in (0, 1/2)$ be fixed and set

$$\ell^2 = 18432\frac{Kc^{1+2\delta}\epsilon^2}{C^4}, \qquad m = \frac{C^2 c^{1-\delta}\log c}{\sqrt{K}\varepsilon}.$$

Suppose that for all $i, j \in [K]$,

$$\mathbb{E}[\text{trace}(A_{i,j}DB_{i,j}(Q-P))] \geq C\,\text{trace}(D)(\text{trace}(Q) - \text{trace}(P))$$

for all permutation matrices D, P, Q with $\text{trace}(P) \leq c - m$, $\text{trace}(D) \geq \ell$, and $\text{trace}(Q) \geq c - m$. Then we have that with probability at least

$$1 - 2K^5 \exp\left\{4c\log c - \frac{C^2 c\log^2 c}{2\epsilon}\right\} - 2K^5 \exp\left\{-c^{1+\delta}\log c\right\}$$

the FW-direction step is maximal in CESSNA for all starting $P = \oplus_i P_i$ with all $\ell \leq \text{trace}(P_i) \leq c - 2m$, and if searching in the j-th component direction, the next iteration of CESSNA will start at a permutation $Q = \oplus_i Q_i$ with $\text{trace}(Q_j) > c - m$.

Proof. Consider searching in component direction j. By Proposition 1.2.1, with probability at least $1 - 2K^4 \exp\left\{-c^{1+\delta}\log c\right\}$, the optimal search direction has trace at least $c - \frac{C^2 c^{1-\delta}\log c}{K\varepsilon}$.

We can find the optimal CESSNA step size in such a direction D' by considering

$$\begin{aligned}
\frac{d}{d\beta}\,\text{tr}\left(AP_{j,\beta}B^T(P_{j,\beta})^T\right)\Big|_1 =& 2\sum_{i\neq j}\text{tr}(A_{i,j}P_iB_{i,j}(P_j - D')^T) \\
&- \text{tr}(A_{jj}P_jB_{jj}(D')^T) - \text{tr}(A_{jj}D'B_{jj}P_j^T) + 2\text{tr}(A_{jj}P_jB_{jj}P_j^T)
\end{aligned}$$

and

$$\begin{aligned}
\frac{d}{d\beta}\,\text{tr}\left(AP_{j,\beta}B^T(P_{j,\beta})^T\right)\Big|_0 =& 2\sum_{i\neq j}\text{tr}(A_{i,j}P_iB_{i,j}(P_j - D')^T) \\
&- 2\text{tr}(A_{jj}D'B_{jj}(D')^T) + \text{tr}(A_{jj}P_jB_{jj}(D')^T) + \text{tr}(A_{jj}D'B_{jj}P_j^T)
\end{aligned}$$

We would like to show all terms in the above derivative expressions are negative (even in the event that some of the P_i's have been updated, though not P_j; as we will show all updated P_i would move to $\Pi_{c,\leq m} \subset \Pi_{c,\leq c-\ell}$ for sufficiently large c). To this end, we consider the event

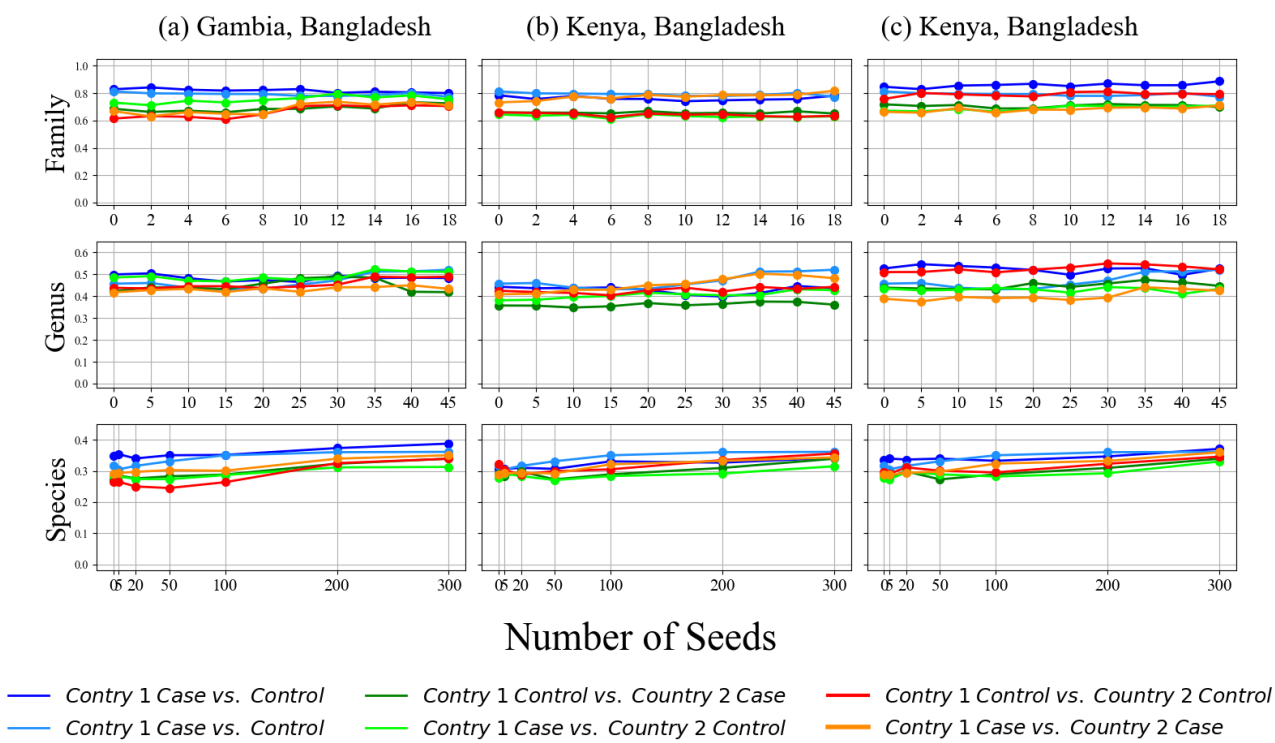
$$\mathcal{E}_1 = \Big\{ \exists Q_1 \in \Pi_{c,\leq m},\, \exists i,j \in [K],\, \exists Q_2 \in \Pi_{c,\leq c-\ell},\, \exists Q_3 \in \Pi_{c,\leq c-\ell} \cap \Pi_{c,\geq 2m}, \\ \text{s.t. } \operatorname{trace}(A_{ij} Q_2 B_{ij} (Q_1 - Q_3)^T) \leq 0 \Big\}$$

A union bounds and application of Lemma 1.1.1 for CorrSBM yielding (upper bounding $d \leq m + \xi$ below) yields that for c sufficeintly large

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_1) &\leq \sum_{\xi=2}^{c-\ell} \sum_{\eta=2m}^{c-\ell} 2K^4 \exp\left\{ (\eta + \xi + m) \log c - \frac{C^2 m^2 (c-\xi)^2}{18432 \epsilon c (m + c - \ell)} \right\} \\
&\leq 2K^4 \exp\left\{ (3c+2) \log c - \frac{C^2 m^2 \ell^2}{18432 \epsilon c (m + c - \ell)} \right\} \\
&= 2K^4 \exp\left\{ 4c \log c - \frac{C^2 c^2 \log^2 c}{\epsilon (m + c - \ell)} \right\} \\
&\leq 2K^4 \exp\left\{ 4c \log c - \frac{C^2 c \log^2 c}{2\epsilon} \right\}
\end{aligned}$$

A final union over search directions, and combining the above probabilities finishes the proof.

2 Graph Matching Ratios Within and Across Countries for Control and Case Groups: country pairs not included in Figure 6



Graph Matching Ratios Within and Across Countries for Control and Case Groups

□

References

[Alon et al., 1997] Alon, N., Kim, J.-H., and Spencer, J. (1997). Nearly perfect matchings in regular simple hypergraphs. *Israel Journal of Mathematics*, 100(1):171–187.

[Fang et al., 2018] Fang, F., Sussman, D. L., and Lyzinski, V. (2018). Tractable graph matching via soft seeding. *arXiv preprint arXiv:1807.09299*.