Inferring and Analyzing the Present and the Past of Networks from Limited Information

Emre Sefer

February 2015

Joint CMU-Pitt Program in Computational Biology
Carnegie Mellon University, School of Computer Science
Pittsburgh, PA 15213 USA

Thesis Committee:
Carl Kingsford - Carnegie Mellon University, Chair
Russell Schwartz - Carnegie Mellon University
Seyoung Kim - Carnegie Mellon University
Chakra Chennubhotla - University of Pittsburgh
Guy Blelloch - Carnegie Mellon University

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Keywords: Network, Diffusion, Combinatorial Optimization, Prediction, Deconvolution

To my family ...

Abstract

Biological networks, social networks and the dynamic processes over them such as diffusion can be better understood by simultaneously analyzing both the network data and the diffusion data. However, data about diffusion, the network, and node attributes are all limited and often wrong. Overcoming this limited/uncertain data bottleneck is an important challenge in better estimating the network structure, better finding the correlations hidden in the network, and better tracking the diffusion dynamics over the network.

We focus on four different problems regarding the analysis of networks and diffusion dynamics over them with limited information. We first improve protein annotation prediction performance by metric labeling and associated semi-metric embedding of the annotations that integrate the similarities between annotations to protein network data. Second, we propose methods to reconstruct an unknown network from available diffusion data accurately at both micro and macro scales in both biological and social domains. Then, we formulate the diffusion history reconstruction problem to estimate the diffusion histories from incomplete snapshots of the diffusion process, and apply our methods to different diffusion types with accurate performance. Lastly, we propose novel methods to deconvolve the biological 3C interaction matrix that is an ensemble over a cell population under several assumptions about their structures. All these problems are computational, and we validate the effectiveness of our methods with both computational experiments and with theoretical bounds.

Acknowledgments

The Research Thesis was done under the supervision of Prof. Carl Kingsford at CMU Computational Biology Department under School of Computer Science.

I would like to thank Carl, my advisor, who wisely led me and guided me throughout the years of my work. He pointed out many interesting new research directions when I got stuck. I would also like to thank him for his great support and encouragement, especially when papers were rejected time after time. I appreciate the good fortune of having the chance to work with him.

I would like to thank my family and research group members for their contributions during the group meetings and discusssion hours. I also thank Geet Duggal and Darya Filippova for their sincere help during my studies. Lastly, I also want to thank to all professors whom I have taken courses during my graduate studies.

The work in this thesis was partially funded by the US National Science Foundation (CCF-1256087, CCF-1053918) and US National Institutes of Health (R21HG006913 and R01HG007104). This research is funded in part by the Gordon and Betty Moore Foundations Data-Driven Discovery Initiative through Grant GBMF4554 to Carl Kingsford.

. Related Publication List:

[1] **Emre Sefer**, Carl Kingsford. Convex Risk Minimization To Infer Networks From Probabilistic Diffusion Data At Multiple Scales. In IEEE International Conference on Data Engineering ICDE 2015.

[2] **Emre Sefer**, Carl Kingsford. Diffusion Archaeology for Diffusion Progression History Reconstruction. in Proceedings of the 2014 IEEE 14th International Conference on Data Mining ICDM 2014. IEEE Computer Society, 2014.

[3] **Emre Sefer**, Geet Duggal, and Carl Kingsford. Deconvolution Of Ensemble Chromatin Interaction Data Reveals The Latent Mixing Structures In Cell Subpopulations. The 19th International Conference on Research in Computational Molecular Biology, RECOMB 2015

[4] Geet Duggal, Rob Patro, **Emre Sefer**, Hao Wang, Darya Filippova, Samir Khuller, and Carl Kingsford. Resolving spatial inconsistencies in chromosome conformation measurements. Algorithms for Molecular Biology, 8(1):8, 2013.

[5] R. Patro, G. Duggal, **E. Sefer**, H. Wang, D. Filippova, and C. Kingsford. The missing models: A data-driven approach for learning how networks grow In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 12, pages 4250, New York, NY, USA, 2012.

[6] **E.Sefer** and C.Kingsford. Metriclabeling and semi-metric embedding for protein annotation prediction. In Proceedings of the 15th Annual International Conference on Research in Computational Molecular Biology, RECOMB11, pages 392-407.

[7] R. Patro, **E. Sefer**, J. Malin, G. Marcais, S. Navlakha, and C. Kingsford. Parsimonious reconstruction of network evolution. Algorithms for Molecular Biology, 7(1):25, 2012.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, increasing availability of technological, sociological, and biological data has aroused considerable interest in developing methods to analyze them in detail and improving the existing data mining tools to infer novel data patterns. In accordance with the increasing data availability, there has been significant interest in graph structures in technological, sociological, and biological settings [9, 10, 36, 90, 108, 110, 148, 154]. In networks, each node models an entity and its associated attributes. For instance, in protein-protein interaction (PPI) networks, each node represents a protein, and each protein has multiple functions in the organism which are its attributes. Similarly, in the Facebook social network, each node represents a human with a Facebook account and there is an edge between two nodes when the two people are friends. These networks may be either unweighted or weighted to represent quality or confidence of the interactions. Additionally, some of the networks may change over time such as a citation network where nodes represent papers and there is a directed edge between two papers if one paper cites the other one.

In many cases, networks are not completely static objects: Even if their structure does not evolve over time, *dynamic processes* occur over them which are affected by both nodes and edges of the graph according to the rules of the process. Ignoring the dynamic process dimension may lead to incomplete analysis of the network. For instance, protein interaction network shows the interactions between proteins, and several cell signaling pathway dynamics also occur over the interaction network. A signaling cascade starts at a protein, and the signal spreads to the proteins at different cell locations via interactions. In this case, proteins directly affect the regulation of the other ones via interactions, and ignoring the signaling dynamics over interaction network will lead to an incomplete picture of cell.

Diffusion is special case of those processes in which a spread (e.g., an infection) starts from some part of the graph and spreads to other portions over time via the edges of the graph [109, 152]. Some examples are virus epidemic on a human-contact network [121], contaminant diffusion over water distribution network [91], and idea spreading over Twitter [83]. A diffusion model defines a set of possible states that the nodes of the graph can be in as well as rules for probabilistically switching between those states. For instance, SIR model is a well-known example of diffusion model that is often used to simulate the spread of influenza between humans. Other widely used diffusion models are SI, SIR, SEIR, SEIRS, SIS, SIRS, etc [61]. These Markovian models are recently brought together under *VPM* (Virus Propagation Model) [112] which

provides a common framework for all those Markovian diffusion models as well as defining hierarchical relationships between them.

Network data and the diffusion data over networks dramatically improved our understanding of both social and biological networks, and the diffusion dynamics over them such as influenza diffusion, opinion diffusion, email virus diffusion [24, 91, 121]. Analyzing both the network and diffusion data simultaneously helps us in better analysis of both the network and the diffusion dynamics. However, network data, data about the node attributes, and the diffusion data are noisy and limited due to several reasons some of which are:

- Network data and its node attributes may only be available partially due to experimental errors. Protein-protein interaction data is an example where it is impossible to correctly measure all protein-protein interactions. Similarly, all protein annotations are not known due to experimental limitations.
- Network data may not be fully available due to privacy. For instance, Facebook does not make the whole friendship dataset publicly available since these datasets might then be used for bad purposes.
- Diffusion data over network may be noisy depending on the way it is collected. When tracking the influenza diffusion over human-contact network, users do not suddenly show the influenza symptoms, and there is a chance of misidentifying the time point when human is infected with influenza. Similarly, gene expression datasets at different time points provide information about the cell signaling, but they are highly noisy especially at higher temporal resolutions.
- Available network data may be noisy since it may have been collected over an ensemble of scenarios representing their average. For instance, Hi-C [94], an experimental method for indirectly measuring the 3 dimensional distances between genomic fragments, collects the interaction data over a population of cells rather than a single cell.

These challenges mainly motivate the problems in this thesis. In this thesis, we propose solutions to four different problems over social networks, biological networks, and diffusion dynamics over them. Common to all these problems, available data is noisy and limited, and we need well-formulated methods with provable performance guarantees. Chapters of this thesis are organized as follows:

Chapter 2 motivates the problems in the thesis by introducing the related network data such as protein-protein interaction data [67], chromosome conformation capture data such as Hi-C [94], and diffusion dynamics over networks such as SI, SIR, SEIR, SEIRS [112].

Chapter 3 proposes a solution for the protein annotation prediction problem where we are interested in predicting the annotations of proteins given protein interaction network and partially known annotation data. This problem is important since approximately 25% of human, and 57% of fly genes do not have known biological functions in the Gene Ontology (without considering the electronically annotated proteins) [29, 84], and annotating them experimentally is highly costly. This problem is important since several functions of the existing proteins are still unknown, and annotating them experimentally is highly costly. Then, it is important to develop fast computational techniques that predict novel annotations by exploting PPI network structure and the similarities between the protein functions. We solve this problem under the realistic assumption that there is a correlation between protein-protein interaction network struc-

ture and the protein annotations: Interacting proteins tend to have similar functions. We propose metric labeling and related semi-metric embedding approaches to integrate network structure and the similarity between the proteins. We show that both problems can be solved optimally, and our methods are scalable to prediction over interaction data of many species. Our methods outperform the existing methods in function prediction which shows the importance of using manually-labeled protein annotations in predicting newer annotations as well as integrating the network structure and the similarity between protein annotations to the prediction framework.

Chapter 4 proposes a method *CORMIN* to infer the unknown graphs at multiple scales from multiple noisy diffusion data. This problem is important when it is easier or less costly to observe the states of the nodes than it is to observe the edges of the network over which the diffusion process is spreading. In a similar case, we are also interested in understanding the diffusion characteristics at the macroscale since it is infeasible and unnecessary to learn it on micro level (person-to-person contact). We model this problem as an expected loss minimization problem where the diffusion data may also be noisy, under-sampled or unobserved. We prove that this problem can be solved optimally for reconstructing the networks at both micro and macroscales. We validate the network reconstruction performance over human-contact network at microscale and estimate the influenza transmission rates between U.S. states at macroscale by using Google Flu Trends data which would otherwise have been impossible at that scale. Our improved formulation leads to a better performance in almost all realistic test cases.

Chapter 5 considers the problem of reconstructing the diffusion history from present-day diffusion data even though we may not know much about their histories. This problem is important in real-life situations since it is not always easy to know the whole diffusion progression, initial diffusion conditions, or the time it has started due to several limitations. It is invaluable to learn more about the past to take precautions to prevent future epidemics, to learn more about the true diffusion mechanics, to guide the behaviour of the diffusion via incentivization, etc. We formulate this problem as a maximum likelihood estimation for SEIRS type models, discuss the hardness of the problem for different types of SEIRS models, and develop various methods to reconstruct histories provably suboptimally. For larger networks, we also develop relaxation methods to reconstruct the diffusion histories over very large networks with provable performance guarantees. We validate the performance of all our methods (*DHR-sub*, *DHR-pcdsvc*, *DHR-pcvc*) by identifying the initial contamination sites over a water distribution network, and by reconstructing the meme diffusion history over a blog network. All our methods can accurately reconstruct the diffusion histories, and predict the initial spreaders over multiple networks.

Chapter 6 considers the problem of deconvolving chromosome conformation capture (3C) interaction data collected over a population of cells. Under several realistic assumptions about the convolved populations, we present a variety of algorithms to deconvolve these measured interaction matrices into estimations of the contact matrices for each subpopulation of cells and relative densities of each subpopulation provably suboptimally. We evaluate the performance of our methods in deconvolving the ensemble interaction matrix on HeLa, mouse, and bacteria data. Our methods are the first methods for 3C deconvolution, and they outperform all reasonable baselines as well as running in less than 15 minutes on almost all datasets on a personal laptop. We also show that domain boundaries from deconvolved matrices are often more enriched or depleted for regulatory chromatin markers when compared to boundaries from the convolved matrices.

Lastly, conclusion chapter 7 summarizes our contributions in this thesis, and discusses several possible future directions. In summary, we show that improved modeling leads to methods that can accurately infer the missing or latent diffusion and network data in social and biological networks. In all problems, our estimates are quite helpful in answering the following previously unanswered questions: (1)- How efficiently can protein annotations be predicted by integrating Gene Ontology data?, (2)- How efficiently can networks be inferred from diffusion data at multiple scales?, (3)- Can we reconstruct diffusion histories accurately over different types of networks?, and (4)- Can we infer latent mixing interaction matrices from ensemble Hi-C data over cell populations?

Problems considered in this thesis are different than the existing work in several ways. Completing the partial knowledge in biological networks has been previously discussed for different types of annotations [34, 43]. In the few cases it has been done, integrating Gene Ontology knowledge into protein function prediction methods [12, 34] and clustering [25] has resulted in improved predictions. However, these methods are in general less systematic and cannot use the manually-curated Gene Ontology hierarchy [5] as well as our approaches. In other cases, the problem of inferring the unknown graph structure from diffusion data has also been previously considered, but most of the existing methods make a homogenous network assumption by ignoring the effect of the network structure in diffusion. These methods neglect the possibility of partially observable, under-sampled probabilistic diffusion data, and they cannot model the uncertainty inherent in the diffusion data. Another shortcoming of the existing approaches is their inability to estimate the diffusion rates at the macroscale.

On the other hand, complete diffusion history reconstruction has not been previously studied but similar problems exist in the literature. The most relevant such problem is *Initial Spreader Identification* where we want to identify the most probable initially infected nodes that started a diffusion. Existing methods for this problem are either based on a heuristic or they only work on restricted set of diffusion models. None of these methods infer the whole diffusion progression, as our approaches do. Lastly, the 3C deconvolution problem has not been previously studied, and all the existing methods on 3C data work on population of cells neglecting the fact that it is an ensemble over cells with different structures. Detailed contributions of our methods are discussed in the relevant sections.

Chapter 2

Preliminaries

2.1. Protein-protein Interaction Networks

There are variety of experimental methods to measure the interactions between proteins. Some examples are yeast two-hybrid [139], chromatin immunoprecipitation [28], tandem affinity purification [114], bimolecular fluorescence complementation [74]. The result of each protein interaction experiment is binary interaction graph $G = (P, E)$ over all considered proteins P. These high-throughput methods have different false positive and false negative rates, and multiple experiments can be combined into a single graph to obtain a more robust estimate of interaction graph.

Protein-protein interaction (PPI) networks have been significantly used in protein annotation prediction. The intuition is that pairs of proteins that are highly related (tend to interact significantly) ought to be assigned labels that are highly similar. Figure 2.1 illustrates the part of Saccharomyces cerevisiae (yeast) interaction network where different colors represent different annotations, and interacting proteins tend to share similar colors. Graph-based representation of protein interaction network makes it easier to apply variety of prediction algorithms over graphs to protein annotation prediction over PPI. In order to obtain a more detailed explanation of protein interaction networks, see the relevant papers [28, 68, 114].

2.2. SEIRS Diffusion Dynamics

SEIRS (Susceptible Exposed Infected Recovered Susceptible) diffusion dynamics over directed graph $G = (V, E)$ with possible state transitions are shown in Figure 2.2. The SEIRS states are Susceptible (*S*), Exposed but not contagious (*E*), Infected and contagious (*I*), and previously infected but now Recovered (or immune to the infection) (*R*). Those states are general enough abstractions to model various forms of diffusion in different contexts [91, 121]. For instance, the infected state can model people having influenza symptoms in influenza diffusion over humans, and it can represent the creation of a blog entry about a topic in idea diffusion. Similarly, the recovered state could represent recovery of a person from influenza or the decontamination of a water tower from chemical contaminants depending on the context.

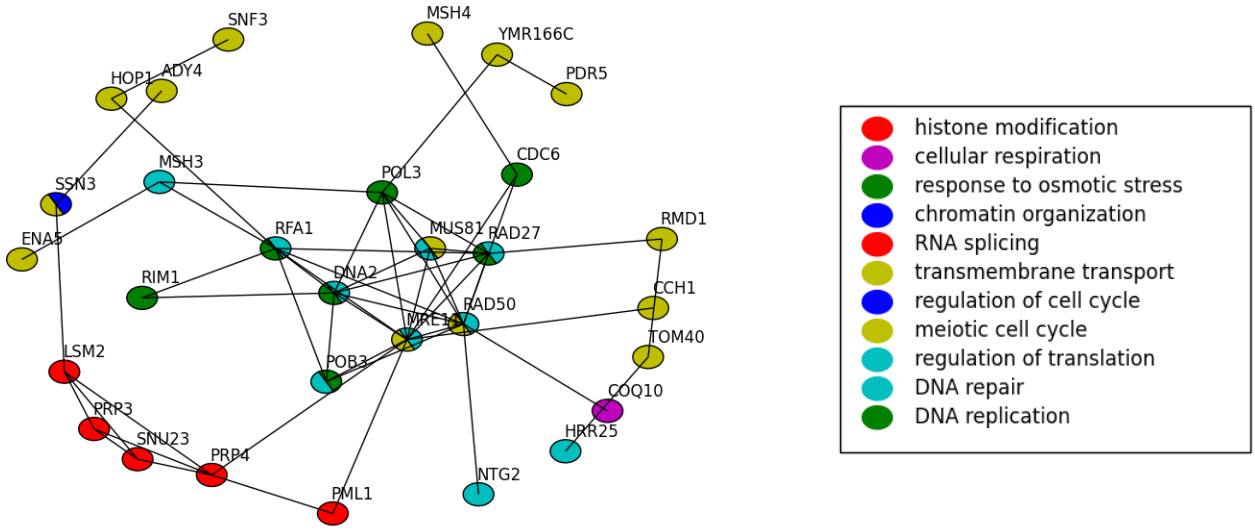


Figure 2.1: Part of yeast protein interaction network (Generated according to Gene Ontology annotations)

In SEIRS model, diffusion starts at time $t = 0$ from set of initially infected nodes and progresses over G in discrete time steps. Let S_t, E_t, I_t, R_t be the set of *S*, *E*, *I*, *R* nodes at time t respectively. At each time step, infected nodes spread the infection to the susceptible nodes with certain probability. This $S \to E$ transition is exogenous; it is affected by G and probability of exogenous transition for susceptible node v at time t is $1 - \prod_{u \in P(v) \cap I_t}(1 - p_{uv})$, where $P(v)$ is the set of nodes with edges into v and p_{uv} is the probability of transmission of the agent over edge (u, v). The remaining $E \to I$, $I \to R$, $R \to S$ transitions are endogenous; their transition probabilities are $e2i_v$, $i2r_v$, $r2s_v$ respectively, and they are not affected by G. For every node at each time step, if a transition succeeds, the node transitions to a new state. Otherwise, it follows similar procedure at next time step, independent of the previous trials. SEIRS type models are

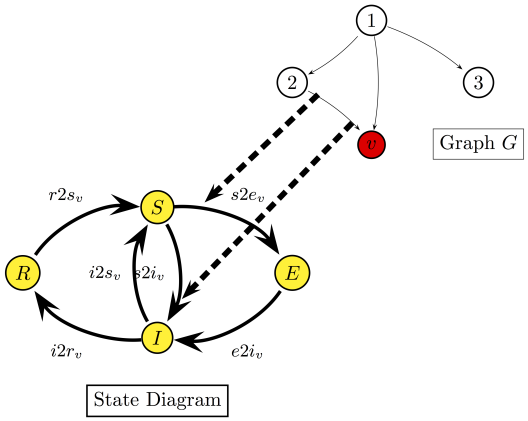


Figure 2.2: SEIRS state transition diagram

Markovian since state of a node at time t depends on its state and its neighbors' states at previous time steps, and it obeys independent cascade (IC) [73] assumption which states that a diffusion from one of nodes predecessor is enough for node to become exposed/infected. Overall, problems over these dynamics will be mostly submodular maximization or minimization [73]. Here, we present transition probabilities same for each time step to simplify the explanation. They may be different as well to model arbitrary distributions, and we will explain this case in more detail in the following chapters.

SEIRS-type models include the well-known SI, SIR, SIS, SIRS, SEIR, SEIRS models [61]. SEIRS is the most general model among these models, and some of its transitions disappear or change slightly in other models. For instance, in SIR, there is no exposed state; the exogenous transition is $S \rightarrow I$ since nodes proceed directly to the infected state, and there is no $R \rightarrow S$ transition. We can classify SEIRS type models in various ways. SIRS, SEIRS are *loopy* models where $R \rightarrow S$ transition is available whereas SI, SIR, SEIR are *non-loopy* models.

2.3. Chromosome Conformation Capture (3C) Experiments

There are two main experimental techniques to understand 3D genome shape. Microscopy imaging techniques, such as Fluorescence In Situ Hybridization (FISH) [92], are medium to low resolution. We may not observe the whole 3D genome shape in greater detail by them. In contrast, high-throughput chromosome conformation capture (3C) based methods such as 3C, 4C, 5C, Hi-C, TCC [39, 71, 94, 137] analyze the 3D organization of chromosomes at a resolution higher than the microscope experiments [39]. 3C-based techniques result in a matrix of counts representing the frequency of cross-linking between the restriction fragments of DNA that are measured over millions of cells. Different chromosome conformation methods differ mainly in terms of their scales. For instance, 5C returns the interaction matrix inside a single chromosome whereas Hi-C measures the interactions between the restriction sites at a genome-wide scale.

The ability to analyze the organization of chromosomes at a genome-wide scale is the main advantage of the high-throughput 3C techniques over the microscope experiments. By means of 3C-based methods, we observe novel long-range interactions between distant genomic loci belonging to different gene clusters [94] which improves our understanding of the cell dynamics at a genome-wide scale. On the other hand, 3C interaction data is collected over a population of cells with variety of genome shapes that are due to temporal and spatial factors. Temporally, cells are in different stages of the cell cycle expressing different sets of genes during the experiment. These expression differences result in different 3D genome shapes [32], and as a result, lead to different 3C interaction matrices. Spatially, each cell in the population may show different rate of cellular stress response to the environmental stressors as well as being affected by the differently distributed epigenetic factors such as histone methylations [11]. Different rates of cellular response result in different expression profiles, and different histone densities affect the probability of interaction between genomic loci [35] both of which in turn affect the genome shape. For instance, barrier insulators that are enriched for certain histone methylations prevent the spread of heterochromatin from a silenced gene to an actively transcribing gene [11].

3C-based methods measure the interactions between the restriction sites over a population of cells as described in [94]. After the basic experimental procedure, our region of interest,

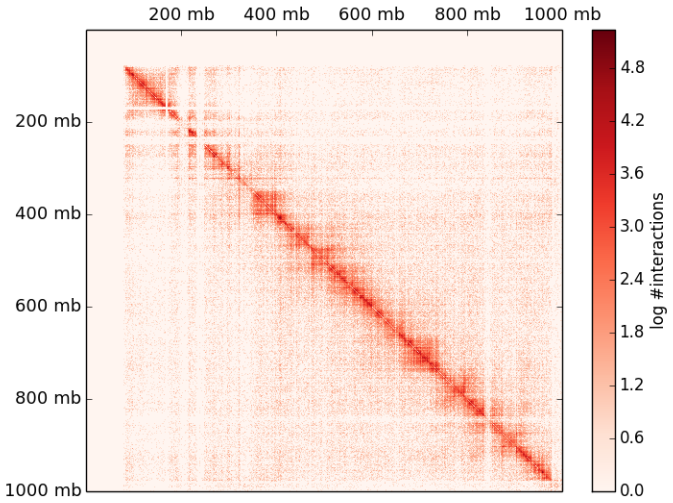


Figure 2.3: Mouse cortex chromosome 19 Hi-C matrix at 40 kb resolution from Dixon et al. [35] (only the first 1000 mb portion is shown).

such as a single chromosome, is divided into equal size bins at a given resolution. Restriction sites are then mapped to their corrresponding bins, and 3C data is turned into a matrix of counts $\mathbf{F} : V \times V \rightarrow \mathbb{R}_0^+$ where $V = \{1, 2, \ldots, n\}$ is the set of bins, and $F_{u,v}$ is number of interactions between all pairs of restriction sites mapping to the bins u and v. For instance, Figure 2.3 shows $\mathbf{F}$ for Hi-C data of mouse cortex chromosome 19 binned at 40 kb resolution in logarithmic scale. In 3C-based experiments, $\mathbf{F}$ shows the interaction counts aggregated over a cell population, however, we do not know the details of which interaction appeared in which cell at how many times. In contrast to the microscopy experiments, 3C methods can estimate the genome shape and the associated features at a higher scale. Even though 3C experiments do not provide the explicit distances between genomic loci, resulting frequencies are inversely correlated with the actual distances. These frequencies must be further post-processed to remove the experimental biases. Then, 3D genome structure is estimated indirectly by mapping the normalized frequencies to 3D distances under an optimization framework such as multidimensional scaling [146].

Estimated genome structure provides us novel insights about the genome. For instance, mammalian genomes at higher resolution show highly interacting regions that are closely embedded in 3D. These regions are called topologically-associated domains (TADs), and TAD boundaries are enriched for histone methylation markers H3K4me3 and H3K27ac, and they are depleted for H3K9me3. Some of these markers have insulator roles, and they have critical roles in genome 3D shape formation [35]. Recently, it was also observed that eQTLs are statistically significantly closer in 3D to their regulated genes than expected by chance [40]. Expression in the beta-globin locus is mediated by folding to bring an enhancer and associated transcription factors within close proximity of a gene [13, 142]. Modeling the three-dimensional shape of genome is thus essential to obtain a more complex understanding of the cell dynamics.

Chapter 3

Predicting Protein Functions By Metric Labeling and Semi-metric Embedding

A preliminary version of this chapter appeared in Research in Computational Molecular Biology - 15th Annual International Conference RECOMB 2011 with the title *Metric labeling and semi-metric embedding for protein annotation prediction* [126].

3.1. Introduction

Networks encoding pairwise relationships between proteins have been widely used for protein function prediction and for data aggregation and visualization. Sometimes these networks are derived from a single data source such as protein-protein interactions [67, 115, 144]. In other instances, they are constructed from integration of large collection of experiments involving different data types, such as gene expression [52], protein localization [66], etc. The precise meaning of an edge can differ, but a common feature of these networks is that two proteins connected by an edge often have similar functions. By extension, these networks generally have the property that two proteins that are "close" in the network are more likely to have closely related functions. This correlation has given rise to a number of computational approaches to extract hypotheses for protein function from relational data [34, 62, 69, 103, 125, 134].

Nearly all of these computational methods treat the function prediction problem as a labeling problem, where the labels are drawn from a vocabulary of biological functions or processes. They typically ignore any relationships between the functions, treating them as independent labels. However, there are usually known relationships among functions that ought to be useful to make more accurate predictions of protein function. For example, the Gene Ontology (GO) [5] is a manually curated database of biological functions and processes that represents the hierarchical relationships among different functions as a DAG. Part of hierarchical GO structure is visualized via AmiGO [22] as in Figure 3.1. However, most prediction methods have ignored such a structure.

In the few cases it has been done, integrating Gene Ontology knowledge into protein function prediction methods [12, 34] and clustering [25] has resulted in improved predictions. For example, Barutcuoglu et al. [12] developed a Bayesian framework for combining multiple SVM

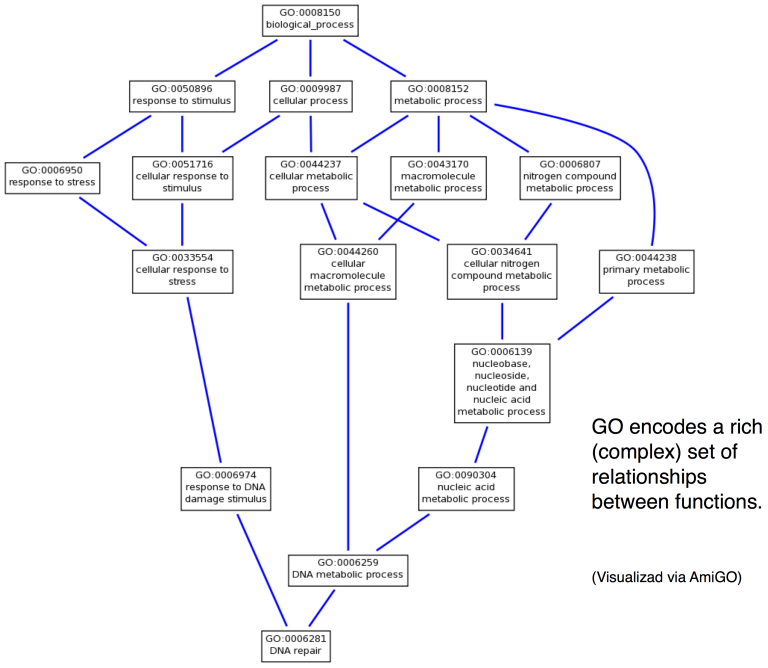


Figure 3.1: Part of Gene Ontology hierarchy

classifiers based on the GO constraints to obtain the most probable, consistent set of predictions. Their approach used a hierarchy of support vector machine (SVM) classifiers trained on multiple data types. This method also exploits the relationship between functions in GO but does not exploit distances between functions directly. Taking another approach, Deng et al. [34] uses the correlations between which proteins are labeled with each functions but they estimate these correlations from training data and do not consider GO structure.

3.1.1 Metric Labeling for Function Prediction

Here, we propose to integrate Gene Ontology relationships with relational data by modeling the protein function prediction problem as an instance of METRIC LABELING [76] which is a special case of MRF [79] in which the distance function among labels is a metric. The METRIC LABELING problem seeks to assign labels (here, protein functions) to nodes in a graph (here, proteins or genes) to minimize the distance (in the metric) between labels assigned to adjacent nodes in addition to the cost of assigning labels to nodes. The advantage of this formulation is that rather than treating function labels as independent, unrelated entities, their similarities can be directly incorporated into the objective function. A more detailed description of the METRIC LABELING problem is given in Section 3.2.1.

The METRIC LABELING formulation can be seen as an generalization of minimum multiway cut [149], which implicitly assigns distance 0 between two identical functions and distance 1 between any pair of distinct functions. METRIC LABELING softens this to account for varied levels of similarities between the functions. METRIC LABELING can also be seen as special case of Markov Random Field (MRF). MRFs encode the same combinatorial problem, but

the distance function is not restricted to metrics or semimetrics [79]. However, optimization with such arbitrary distance functions is NP-Hard [27, 79], and there is no approximation algorithm that can approximate the global optimum within a non-trivial bound. In contrast, there are practical approximation algorithms for METRIC LABELING with logarithmic approximation guarantees [23, 76]. In this research, we will use the integer programming formulation by [23] which yields an $O(\log k)$ approximation algorithm for METRIC LABELING where k is number of labels.

3.1.2 Constructing a Metric Distance Between GO Functions

METRIC LABELING (and MRF models) have typically been used in applications related to computer vision [18, 79, 93] where often the distance between the labels naturally can be expressed by metrics. In the case of function prediction from relational data, while heuristic relationships between functions can be readily computed from the structure of the Gene Ontology graph, it is more difficult to make these distances obey the requirements of a metric. Recall that a **metric** $d(\cdot, \cdot)$ over items X satisfies the following 4 properties for all x, y, z in X:

$$d(x, y) \geq 0 \qquad \text{(Nonnegativity)} \tag{3.1}$$

$$d(x, y) = 0 \text{ if and only if } x = y \tag{3.2}$$

$$d(x, y) = d(y, x) \qquad \text{(Symmetry)} \tag{3.3}$$

$$d(x, z) \leq d(x, y) + d(y, z) \qquad \text{(Triangle Inequality)} \tag{3.4}$$

Typically, properties (3.1)–(3.3) can be easily satisfied, but often natural distance measures do not satisfy the triangle inequality (3.4). When d satisfies (3.1)-(3.3) but not the triangle inequality (3.4), it becomes a **semimetric**.

To apply METRIC LABELING when the distance function on the labels is merely a semimetric, we will first convert the semimetric into a metric that is as similar to the semimetric as possible. Approximating a semimetric by a close metric and MRF optimization when the distances are semimetric are topics of recent interest and Kumar and Koller [82] have recently suggested an algorithm based on minimizing the distortion. If $\mathcal{S}$ is a semimetric, and $\mathcal{M}$ is a metric approximating $\mathcal{S}$, *contraction* of this mapping is the maximum factor by which distances are shrunk in $\mathcal{M}$ and *expansion* or stretch of this mapping is the maximum factor by which distances are stretched in $\mathcal{M}$. *Distortion* of this approximation is the product of the contraction and the expansion. Although distortion minimization has traditionally been used in metric embeddings, distortion considers the error introduced in the largest outlier and does not take into account the distribution of the error over all the distances. For imperfect data that is far from a metric, intuition indicates that minimizing the error introduced in the other distances would yield a better metric.

To design metric approximations to semimetrics that better preserve all distances, we propose a least-squared minimization algorithm that tries to minimize the total squared error among all distances. To contrast it with traditional distortion, we call this approach *least squared distortion* (LSD). This problem can easily be solved in polynomial time because it is a convex case of quadratic programming. Thus, to apply METRIC LABELING in cases when the distances among the labels are not metric, we first map the semimetric to a close metric using the LSD algorithm

and then run METRIC LABELING on the new metric. Experiments on protein function prediction suggest this is a good metric approximation method. The issue of converting a set of heuristically estimated distances to a metric arises in many practical contexts and the LSD approach may also be useful for other applications.

3.1.3 Improvement in Function Prediction

We test the LSD algorithm and the METRIC LABELING approach for function prediction on relational data for 7 species: *S. cerevisae*, *A. thaliana*, *D. melanogaster*, *M. musculus*, *C. elegans*, *S. pombe* and Human. For *S. cerevisae*, we apply the algorithms to an integrated data set that derives pairwise relationships between proteins from several lines of evidence such as gene expression, protein localization data, and known protein complexes. For all 7 species, we also test the approaches on networks derived from high-throughput protein-protein interaction experiments.

The algorithms are tested in a variety of settings. The set of functional labels are drawn from the Gene Ontology's Biological Process sub-ontology. The number of considered GO terms is varied between 90 and 300 in order to evaluate the effect of the size and specificity of the label set on performance. Specific GO terms are selected for each species to match sets of terms used in previous publications [80] and that species annotation set. Depending on the number of annotations required, those annotations that are seen more than others and also that are not parent of each other are selected. Annotations for each case and for each species can be found on the website. Various metrics and semimetrics relating the GO terms are also tested. A simple shortest-path metric is compared with two other semimetrics derived from lowest common ancestor in the Gene Ontology DAG, semimetrics computed from a training set of labels, and semimetrics computed from both training set and GO. See Section 3.2.4.

3.1.4 Our Contributions

We introduce the use of METRIC LABELING for protein function prediction from relational data and show that under many reasonable metrics it outperforms the approaches based on Markov Random Fields [88], Functional Flow [103], minimum multiway cut [72, 149], neighborhood enrichment [62], and simple majority rule [125]. We test on 7 species in both protein-protein and integrated networks using several different collections of GO terms. The results indicate that the clean METRIC LABELING formulation is useful for automated function prediction.

In addition, we introduce the LSD objective function for finding a metric that approximates a semimetric with the goal of preserving many distances rather than just limiting the maximum distortion. The convex optimization approach for this problem may be useful in other contexts where reasonable heuristic distances do not satisfy the triangle inequality. We compare the performance of running first our LSD metric approximation algorithm and then running METRIC LABELING on the LSD's output metrics with a recent algorithm by Kumar and Koller [82] and METRIC LABELING with LSD metric approximation seems to result in better predictions.

3.2. Methods

3.2.1 The Metric Labeling Problem

The METRIC LABELING problem has been extensively investigated from a theoretical point of view [23, 76]. Formally, we have a graph $G = (P, E)$ over a set P of n nodes (here, proteins), E of edges and a set L of k possible labels (here, functions) that we want to assign to objects. We have a metric $d(\cdot, \cdot)$ satisfying properties (3.1)–(3.4) defined between any labels in L. We are also given a function $c(p, \ell)$ that provides the cost of assigning label $\ell \in L$ to $p \in P$. METRIC LABELING seeks an assignment $f : P \to L$ of labels to proteins that minimizes the objective function:

$$Q(f) = \sum_{p \in P} c(p, f(p)) + \sum_{(p,q) \in E} w(p, q) d(f(p), f(q)). \tag{3.5}$$

where $w(p, q) = w(q, p)$ is the weight of the edge between proteins p and q in the graph. The first summation is called the *assignment costs* and depends only on individual choice of label we make for each protein and second summation is called the *separation costs* and is based on the pair of choices we make for two interacting proteins.

The intuition is that pairs of proteins that are highly related (w_{pq} is high) ought to be assigned labels that are highly similar ($d(f(p), f(q))$ is low). The assignment costs prevent the problem from becoming trivial by forbidding the assignment of the same label to every protein. For a protein p with a known function b, typically $c(p, b)$ will be 0 and $c(p, \ell) = \infty$ for all $\ell \in L$ except b.

3.2.2 Integer Programming Formulation of Metric Labeling

The METRIC LABELING problem defined above can be written as an ILP [23]. In this formulation, $x(u, i)$ is binary variable indicating that vertex u is labeled with i and $x(u, i, v, j)$ is binary variable indicating that vertex u is labeled with i and vertex v is labeled with j for edge $(u, v) \in E$. The objective is then to

$$\text{minimize} \sum_{v \in V} \sum_{i \in L} c(u, i) x(u, i) + \sum_{(u,v) \in E} \sum_{i \in L} \sum_{j \in L} w(u, v) d(i, j) x(u, i, v, j). \tag{3.6}$$

The variables are subject to the following constraints:

$$\sum_{i \in L} x(u, i) = 1 \qquad \forall u \in V \tag{3.7}$$

$$\sum_{j \in L} x(u, i, v, j) = x(u, i) \qquad \forall u \in V, v \in N(u), i \in L \tag{3.8}$$

$$x(u, i, v, j) = x(v, j, u, i) \qquad \forall u, v \in V, i, j \in L \tag{3.9}$$

$$x(u, i) \in \{0, 1\} \qquad \forall u \in V, i \in L \tag{3.10}$$

$$x(u, i, v, j) \in \{0, 1\} \qquad \forall (u, v) \in E, i, j \in L \tag{3.11}$$

Constraints (3.7) mean each vertex must receive some label. Constraints (3.8) force consistency in the edge variables: if $x(u, i) = 1$ and $x(v, j) = 1$, they force $x(u, i, v, j)$ to be 1. Constraints (3.9) express the fact that (u, i, v, j) and (v, j, u, i) refer to the same edge.

Solving this integer programming instance optimally is NP-complete. Since we are dealing with large networks, we use the $O(\log k)$ approximation algorithm given by Chekuri et al. [23] that is based on solving the linear programming relaxation to identify a deterministic HST metric [44] of the given metric such that the cost of our fractional solution on this HST metric is at most $O(\log k)$ times the LP cost on the original metric. We implemented and ran the LP formulation in GLPK [54].

3.2.3 Metric Approximation via Least Square Distortion Minimization

The algorithms suggested above have guaranteed performance bounds when the distance d is a metric. However, finding a metric distance in practical contexts can be difficult. Ideally, the distance encodes a large amount of knowledge about the relationship between protein functions. It is likely that such as distance will not satisfy the triangle inequality. We define a novel metric approximation algorithm, called LSD, based on minimizing the total least squared error between a given semimetric set of distances and the computed metric distances. Least squared error approximation is intuitive because the error of every distance contributes to the total error of the metric approximation instead of only the maximum expansion and contraction as in distortion case.

The LSD algorithm is defined as a quadratic program below, where $S = \{s_1, \ldots, s_{\binom{n}{2}}\}$ is the given set of semimetric distances between each pair of n items, and $M = \{m_1, \ldots, m_{\binom{n}{2}}\}$ is corresponding set of metric distances, where for all i, s_i and d_i represent distances between the same pair of proteins. Let $I = \{1, \ldots, \binom{n}{2}\}$ be the set of indices of distances.

To find a good approximation to the distances in S we seek values for the $\{m_i\}$ variables to

$$\text{minimize} \sum_{i \in I} (s_i - m_i)^2 \quad . \tag{3.12}$$

We require that the m_i values satisfy the following constraints for all $i, j, k \in I$ that should be related by the triangle inequality:

$$m_i + m_j - m_k \geq 0 \tag{3.13}$$
$$m_i + m_k - m_j \geq 0 \tag{3.14}$$
$$m_k + m_j - m_i \geq 0 \tag{3.15}$$

The objective function can be written as $(1/2)x^T Q x + c^T x$ where $n \times n$ coefficient matrix Q is symmetric, c is any $n \times 1$ vector, and x is $n \times 1$ vector of m_i variables. In our case, the matrix Q is positive definite and if the problem has a feasible solution then the global minimizer is unique. In this case, the problem can be solved by interior point methods in polynomial time. We implemented and ran the problem in CPLEX [145].

3.2.4 Metrics and Semimetrics

We test 4 different distance measures between protein functions:

1. $d_{\text{SP}}(x, y) =$ the shortest path distance in the GO DAG between x and y divided by diameter of GO. This is a metric and intuitively simple.
2. $d_{\text{LCA}}(x, y) = (b + c)/(2a + b + c)$, where a is shortest path distance from the root of the ontology to the lowest common ancestor u of x and y and b is the shortest distance from x to u and c is the shortest distance from y to u. The LCA distance measure does not satisfy triangle inequality and is only a semimetric.
3. $d_{\text{Lin}}(x, y) = (\log \Pr(x) + \log \Pr(y))/(2 \log \Pr(\text{lca}(x, y)))$, where $\Pr(x)$ is the empirical probability (computed from the training annotations) that a protein is annotated with x, and lca(x,y) is the LCA of x and y. This is defined in [95] as a similarity measure, and we take its reciprocal as a distance. It is similar to the LCA distance above but uses the probabilities of each annotation instead of GO distances. It has mostly been used in NLP applications [20, 96]. However, it has recently been used in other applications of Gene Ontology distances [37, 123]. It is a semimetric.
4. $d_{\text{KB}}(x, y) = \sum_{p_1 \in P_x} \sum_{p_2 \in P_y} \text{sp}(p_1, p_2)/(\text{diameter} \cdot |P_x| \cdot |P_y|)$, where P_x and P_y are sets of proteins in the training set annotated with x and y respectively, $\text{sp}(x, y)$ is the shortest path distance between x and y, $diameter$ is the diameter of network.

We also consider the combination of a structure-based $d \in \{d_{\text{SP}}, d_{\text{LCA}}, d_{\text{Lin}}\}$ with the knowledge-based d_{KB} using the formula:

$$d_{\text{comb}}(x, y) = (1 - \alpha) d(x, y) + \alpha d_{\text{KB}}(x, y)\,, \tag{3.16}$$

where α is a weight of contribution of training set estimations. For $\alpha < 1$, none of the combined distances are metric (but are semimetric).

When the distance is not a metric, we first run the LSD metric approximation algorithm (Section 3.2.3) to obtain a metric and then run METRIC LABELING on those metric distances. When it is a metric, we just run METRIC LABELING.

In addition, we test two schemes for the assignment costs $c(u, i)$ of assigning label i to node u. Either they are chosen to be uniformly 1 or non-uniformly according to the density of a label in a particular region of the graph as follows: We estimated for each protein p and label i cost $c(p, i) = n_p/(n_{pi} n_p) = 1/n_{pi}$ where n_p and n_{pi} are number of neighbors of p and number of neighbors of p in the training set that have function i respectively. In the case where p has no neighbors with function i, $c(p, i) = 2$. When a function of protein is known, cost of assigning that function is 0 whereas assigning other functions are ∞. Our two-step framework is summarized in Figure 3.2.

3.2.5 Network Data

We tested our algorithm on the protein-protein interaction (PPI) networks of 7 species obtained from BIOGRID [138]: *S. cerevisiae*, *C. elegans*, *D. melanogaster*, *A. thaliana*, *M. musculus*, *H. sapiens*, and *S. pombe*. We used all physical interactions in BIOGRID. Duplicate edges were counted as single edges. We consider only the largest connected component. We used GO annotations downloaded from the Gene Ontology as our true annotations. Only non-EIA annotations are considered. When considering only PPI networks, weight of every edge is 1.

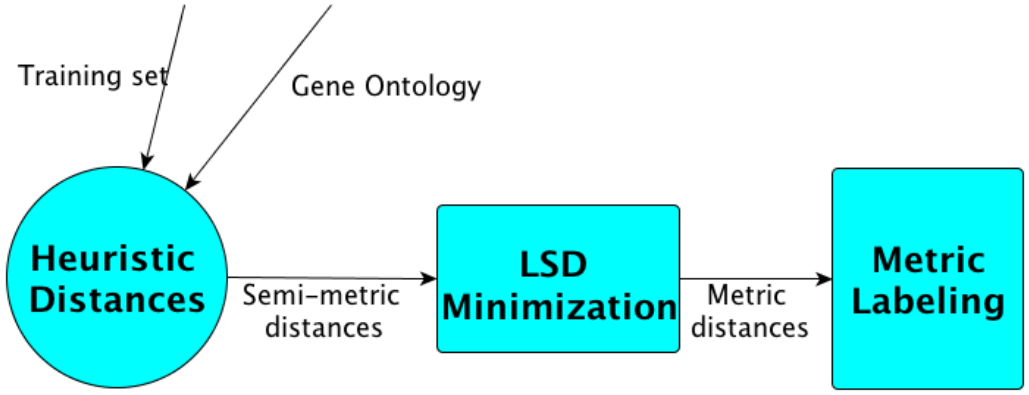


Figure 3.2: Protein annotation prediction framework

For *S. cerevisiae*, we also considered an integrated network derived from several data sources, including gene expression [52], protein localization [66], protein complexes [53, 63], and protein interaction [138]. We used protein complex dataset by assigning binary interactions between any two proteins participating in the same complex, yielding 49313 interactions. For gene expression data, we assigned binary interactions between genes whose correlation in [52] is greater than 0.8 or smaller than -0.8. We assigned binary interactions between any proteins that are annotated to the same location in [66].

We combined these data sources into one network by using noisy-or with their reliability scores, where the interaction score between nodes u and v is taken to be $w(u, v) = \text{Score}(u, v) = 1 - \prod_{i \in E_{uv}}(1 - r_i)$ where E_{uv} are the experiments in which u and v were observed to interact. The reliability r_i of each source i was estimated by the percent of edges from i that connect proteins of shared function in the training data.

3.2.6 Comparison to Other Methods

We run algorithms on a Mac which has 2 GHz Intel Core 2 Duo processor and 2 Gb memory. The METRIC LABELING algorithm took approximately 15 minutes to run. We compared METRIC LABELING predictions with several well-known direct function prediction methods:

Majority: Each protein is annotated with the function that occurs most often among its neighbors as described in [125]. The main disadvantage of this method is that the full topology of network is not considered.

Neighborhood: For each protein, we consider all other proteins within a radius $r = 2$ as described in [62] and a χ^2-test is used to determine if each function is overrepresented.

GenMultiCut: This approach is described in [149] and [72]. It tries to cluster the network by minimizing the number of edges between clusters. This algorithm is a simpler version of our algorithm in which distance between two functions are either 1 (if they are not the same) or 0 (if they are equal). Hence, it cannot take the relations among functions into account. We followed the same approach as [103] and ran an ILP formulation for this

problem 50 times, each time perturbing the weights by a very small offset drawing from uniform distribution on $(-w_{\max}10^{-5}, w_{\max}10^{-5})$ where $w_{\max}$ is the maximum edge weight in the graph. Then probability of assigning a function to a protein will be the fraction of number of annotations of this protein with that function. We implemented this by using MathProg and GLPK. It runs in < 1 minute on yeast.

FunctionalFlow: Each function is independently flowed through the whole network according to an update rule and each node is assigned to functions depending on the amount of flow it receives [103].

MRF: This method is from [88]. It is based on kernel logistic regression which is the improvement over previous MRF models [34, 81]. This method also tries to exploit the relation between different functions by identifying a set of functions that are correlated with the function of interest. However, it does not use the structure of GO when estimating the correlation. This approach takes < 5 minutes to run on yeast.

We also compared LSD with a recent approach for MAP estimation under a semimetric:

Semimetric MAP Estimation Algorithm: This algorithm from [82] tries to approximate a given semimetric distance function using a mixture of r-hierarchically well-separated tree (r-HST) metrics [44]. Then, it solves each resulting r-HST metric labeling problem. We followed the same approach as in GenMultiCut, run the formulation 50 times by perturbing the edges and assign the fraction of number of annotations of this protein with that function as probability of annotating this protein with that function. We modified code provided by the authors to work on our data sets. It ran in less than a 1 minute on yeast.

Solving LSD optimally takes an hour to three hours depending on number of elements in the ontology. However, we only run that once to come up with metrics. This time can easily be reduced to several minutes by considering an iterative approach that starts with point set which elements satisfy triangle inequality and adding other points iteratively by minimizing the total distance modifications made so that current set of points after each iteration will keep satisfying triangle inequality. However, solution of this iterative approach is not guaranteed to be optimal anymore.

3.2.7 Evaluating Performance

We use fivefold cross-validation to compare the predictive performance of the algorithms. The d_{KB} distance and the non-uniform assignment costs are computed using only the remaining 80% of annotated proteins each time. All performance measurements are the average of the 5 runs. Each method described in Section 3.2.6 yields a score, and we assess performance at different false positive rates by varying the score thresholds from 0 to 1 by 0.05 increments. We varied the number of considered functions from 90 to 300. We counted each annotation seperately as a separate example.

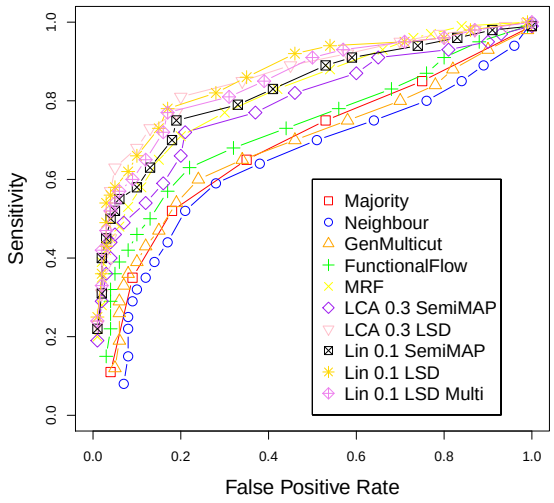


Figure 3.3: ROC curves comparing various algorithms with METRIC LABELING approaches using 90 ontology terms. **SemiMap** indicates the semimetric-to-metric conversation algorithm by Kumar and Koller [82] is run; **LSD** means we first run our LSD minimization algorithm and then METRIC LABELING. The trade-off α between the GO-based distance and the training distance (Equ. 3.16) is either 0.1 or 0.3 as indicated.

3.3. Results

3.3.1 Function Prediction in Yeast Using a PPI Network

Predictive performance on the yeast PPI network is shown in Figure 3.3. The curves show that METRIC LABELING combined with our LSD metric approximation algorithm performs better than the other tested algorithms. METRIC LABELING is more accurate than GenMultiCut in every case since GenMultiCut ignores the effect of distances between functions. FunctionalFlow also does not perform as well as METRIC LABELING, which again may be due to its independence assumption between functions. METRIC LABELING still performs well when number of elements in ontology is 150 and 300 (Figure 3.4).

METRIC LABELING also outperforms the MRF-based algorithm [88]. This may be because the correlation estimations between functions used in that approach depend solely on training data whereas our distances are estimated from both the training set and the structure of the GO DAG. This indicates that, while the Gene Ontology is an imperfect, incomplete, manually edited resource, the distances between annotations in the ontology do contain useful information that can be exploited to make more accurate predictions.

Among various distance heuristics we used, the LCA and Lin distances are better in general since they take the lowest common ancestor into account. The d_{Lin} and d_{LCA} distances perform about the same but they both perform better than the d_{SP} metric (Figure 3.5a). This further indicates that lowest common ancestor is a good distance estimator when there are hierarchical relations among points as shown previously in WordNet [49]. This also echos results in several

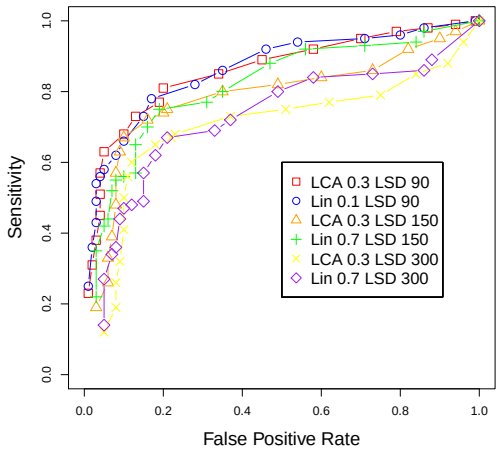


Figure 3.4: Performance of METRIC LABELING degrades as the number of terms increases.

other papers [20, 96, 116] in terms of showing effectiveness of lowest common ancestor as a measure between ontology terms. In addition, in almost all cases the nonuniform assignment costs performs slightly better than uniform assignment costs, although the effect is not large, and if nonuniform assignment costs are not available, uniform assignment costs can be nearly as effective.

Running the LSD minimization for semimetrics and then running METRIC LABELING performs better than Semimetric MAP Estimation algorithm [82] on most of the cases which shows optimizing least squared error, rather than the classical distortion, for metric approximation also seems to be effective in the protein function prediction application.

3.3.2 Trade-off Between GO-distances and Network Distances

We also investigate how performance varies as the tradeoff between a distance computed from the GO structure ($d_{\text{SP}}, d_{\text{LCA}}, d_{\text{Lin}}$) and a distance computed from proximity in the network (d_{KB}) is varied. Figure 3.5b shows the performance of METRIC LABELING with LSD metric approximation and the LCA distance for different trade-offs α between the GO-based structural distance (d_{LCA}) and the trained distances d_{KB} as described in Equ. 3.16. In almost all cases, using distances based solely on GO performs better than using only d_{KB} but using estimations both from training set and Gene Ontology structure performs better than using either one alone.

Combining the Gene Ontology knowledge with training set estimations using low values of α ($\alpha = 0.1$ or $\alpha = 0.3$) achieves the best performance by a slight margin for most of the cases. When the number of elements in ontology increases, best performance is achieved by running METRIC LABELING with combination of d_{Lin} distances and training set estimates when $\alpha = 0.7$. After the initial benefit of using some of the d_{KB} distances, the performance starts to decrease as the weight α is increased. This may mean that d_{KB} is most effective when it operates as a tie-breaker between terms that have the same GO distances. (The dependence on α of the performance of the other GO-based distances d_{SP}, d_{Lin} is similar.)

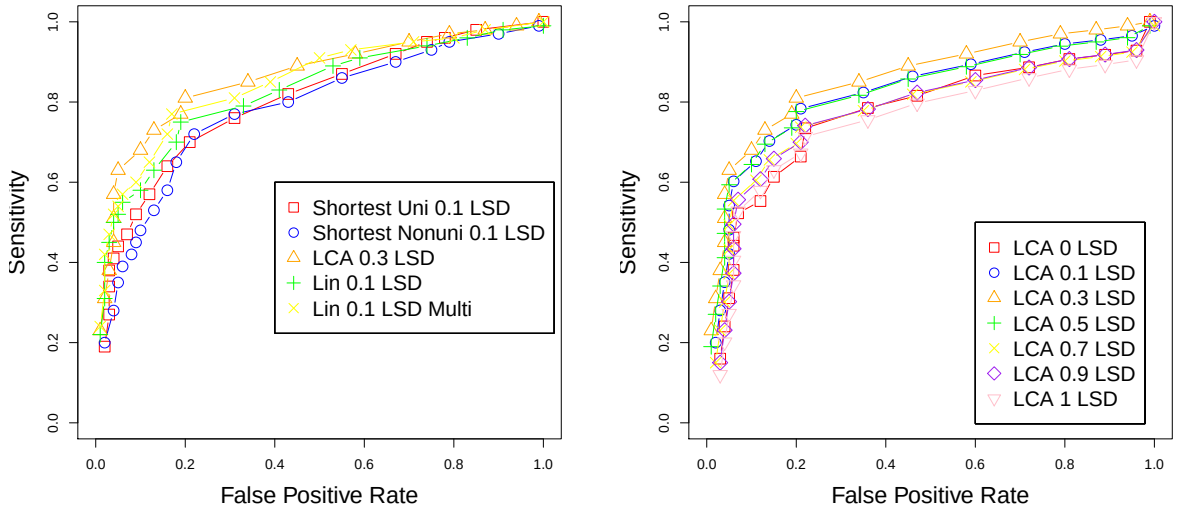


Figure 3.5: (a) Performance of METRIC LABELING combined with the LSD minimization using various distance measures. **Uni** indicates the assignment costs of METRIC LABELING are uniformly 1 except for known annotations and **Nonuni** means assignment costs are nonuniform as described in Section 3.2.4. **Shortest**, **LCA**, **Lin** indicate the different distances functions in Section 3.2.4. (b) Performance of the d_{LCA} distance combined with the d_{KB} distance with various α using the LSD algorithm.

3.3.3 Robustness on the Yeast PPI Network

METRIC LABELING combined with LSD metric approximation is more robust to noise in both misannotations and edge removal as shown in Figure 3.6. We tested for robustness of the predicted results in two ways. First, we removed various percentages of edges randomly from the PPI network and re-run our algorithm. Performance clearly decreased but even when 50% and 40% of the PPI edges are removed on 90 and 150 element ontologies respectively a METRIC LABELING approach performs as well as other algorithms run on the true PPI network. The fewer elements the ontology has, the more robust it is in terms of edge removal. The Lin and LCA distance measures again outperform shortest path distance and running LSD minimization for semimetrics and then METRIC LABELING does better than using the Semi-Metric MAP Estimation algorithm [82]. The LSD minimization may handle the noise in the data better during its error minimization.

Secondly, we also tested robustness by misannotating various percentages of protein annotations and then running our algorithm. Performance even when 30% of the proteins are misannotated on both 90 and 150 element ontologies is still comparable with its performance with the true labels, and it is not worse than other algorithms on the true labels. However, in the case of misannotations, combining GO knowledge with training set estimations ($\alpha = 0.1$ or $\alpha = 0.3$) no longer performs the best. Rather, the GO structure-based distances in isolation perform the best as expected.

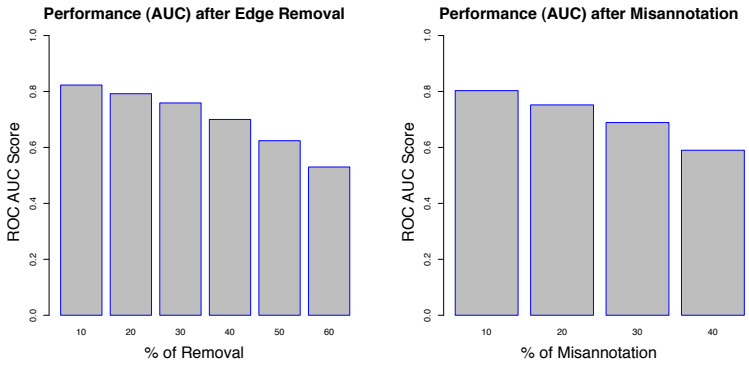


Figure 3.6: Robustness of METRIC LABELING combined with LSD

3.3.4 Performance on Other Networks

When we created integrated network from multiple sources as described in Section 3.2.5, the performance increases slightly (last curve in Figure 3.3). This shows that the METRIC LABELING approach is also useful on relational data other than PPI networks. We also tested our algorithm on several species. Among those species, performance strongly depends on how complete PPI network is, with sparser networks generally exhibiting worse performance. Again, the METRIC LABELING approach performs competitively with existing methods.

3.4. Conclusions

We show that GO structural information can be exploited to achieve better protein function prediction. We also show that the clean, combinatorial problem of METRIC LABELING can effectively use these distances and produce accurate predictions in a reasonable amount of computational time.

Our novel LSD metric approximation algorithm combined with METRIC LABELING performs better than the semimetric MAP estimation algorithm in most cases. This is interesting since distortion defined as in Section 3.1 has nearly always been used as the performance measure for metric embeddings. However, as mentioned, distortion does not consider the distribution of the error on all points. Its minimization considers just the minimization of the boundary cases (of maximum contraction and expansion). LSD minimization instead tries to minimize the total least squared error which makes sense both intuitively and experimentally as we have seen on protein function prediction. Its effectiveness on different application domains is an open question, but the LSD approach is likely to be useful for the common problem of converting a set of heuristic distances into a metric for subsequent processing with an algorithm (such as that for METRIC LABELING) that assume a metric. The LSD metric approximation is completely independent of METRIC LABELING. Either of these algorithms can be changed without affecting the other. However, this is not the case for Semimetric MAP Estimation algorithm, for which the two phases of metric estimation and prediction are not independent.

Chapter 4

Convex Risk Minimization To Infer Networks From Probabilistic Diffusion Data At Multiple Scales

A preliminary version of this chapter appeared in IEEE International Conference on Data Engineering ICDE 2015 with the title *Convex Risk Minimization to Infer Networks from Probabilistic Diffusion Data at Multiple Scales* [128].

4.1. Introduction

Networks are being heavily used to model and analyze the properties of various social and biological systems. The phenomenon of study is often modeled as a dynamic process spreading over the network. Diffusion is special case of those processes in which a spread (e.g., an infection) starts from some part of the graph and spreads to other portions over time via the edges of the graph. Some examples are virus spreading [121], and idea spreading over Twitter [83]. A diffusion model defines a set of possible states that the nodes of the graph can be in as well as rules for probabilistically switching between those states. SEIR [61], for example, is a well-known example of a diffusion model that is often used to simulate the spread of infection. Other widely-used SI, SIS, SIR models [61] are special cases of the SEIR model.

In many situations, it is easier or less costly to observe the states of the nodes than it is to observe the edges of the network over which the diffusion process is spreading. For instance, we might easily observe opinion diffusion on social networks but it may not be possible to see the network due to privacy. Similarly, it is difficult to measure the human contact network for flu transmission [121] but it is easier to detect whether people are ill. In other cases, we are also interested in understanding the diffusion characteristics at macroscale since it is infeasible and unnecessary to learn it on micro level (person-to-person contact). For instance, we are interested in estimating the rates of influenza diffusion between the U.S. states but not on the person to person details of this transmission. In this research, we study the problem of inferring the unknown network when all we are able to observe are traces of how the states of node change as the diffusion spreads over the graph. In terms of influenza diffusion, unknown network models the

contacts between humans at microscale, whereas it represents influenza diffusion rates between U.S. states at macroscale. Recovery of the transmission network is important in designing better epidemic containment strategies and better vaccination strategies.

We present *CORMIN* (COnvex Risk Minimization to Infer Networks) that addresses the problem of inferring the graph from the diffusion data in less idealized and more applicable settings. First, we explore the case that diffusion data is not perfectly known. This uncertainty in the diffusion data is interpreted differently in different contexts. For instance, when tracking the spread of a disease, measured symptoms such as headache and fatigue only partially reveal a node's state since they are not perfect representatives of diffusion states (infected, etc.). Further, the infected person does not suddenly start showing all the symptoms but instead severity of the symptoms increase progressively over time. In this case, we cannot perfectly know the diffusion times but rather estimate our degree of belief (confidence) of being at certain states. When estimating the influenza diffusion rates between the U.S. states at macroscale, probabilistic modeling is mandatory since the diffusion data is an ensemble over many people, and probabilistic data in each U.S. state is interpreted as the percentage of people infected with influenza in that U.S. state. Second, obtaining diffusion data is often expensive, so we may not know status of nodes at each possible time step but rather observe them with frequency lower than that at which the diffusion model is operating. Lastly, we infer networks from SEIR model and its special cases at both micro and macro scales.

Our main innovation to tackle these challenges is to treat diffusion data for each node and each possible state as probabilistic time series. This is in contrast to the existing diffusion-based inference methods [55, 56, 58, 102, 117] for which a node is in each state with either probability 0 or 1. We formulate the graph inference problem as L1 regularized risk (expected loss) minimization program from SEIR dynamics. When the diffusion data is perfect, L1 regularization can be removed and *CORMIN* can be run nonparametrically by adding constraints that force at least a single edge to exist between a newly infected node and the previously infected nodes that are not yet recovered. We applied *CORMIN* to infer synthetic networks, high school human contact network at microscale, and to estimate influenza diffusion rates between U.S. states at macroscale.

CORMIN is capable of inferring the graphs under many challenging cases, and we found it to perform consistently better than the existing methods in almost all cases due to its probabilistic formulation even though we run the competing methods with their best parameters. Performance of *CORMIN* is not significantly affected by the probabilistic data whereas the existing methods performance decreases even though we apply a non-naive rounding scheme to pre-process the input to make schemes designed for 0/1 probabilities work with more general probabilities. For instance, *CORMIN* can achieve $F0.1$ score around 0.7-0.8 over a human contact network if the traces are the only prior information available about the graph. It can also nicely model and infer the influenza diffusion between U.S. states at macroscale that cannot be done by the existing methods. At macroscale, we found the influenza transmission rates between U.S. states estimated by *CORMIN* on Google Flu Trends dataset to be correlated with the human transportation rates between those states. Estimated diffusion rates between U.S. states are asymmetric, and the diffusion rates between less populous states are high especially when they are close to each other.

In summary, probabilistic modeling of the observed data, and the ability to model both edge

existence and non-existence is the main reason *CORMIN* outperforms the other methods on both real and synthetic data under various challenging cases. In contrast to the existing methods, we may also use *CORMIN* to estimate the diffusion rates at macroscale via its probabilistic formulation. *CORMIN* still performs reasonably well when the noise dynamics parameters that map exact transition times to the observed diffusion data are also unknown. In this case, it can simultaneously estimate the noise dynamics parameters and infer the graph which cannot be done by the existing methods.

4.1.1 Related Work

Many existing methods [3, 7, 61] model the influenza transmission by differential equations; they make a homogenous network assumption by ignoring the effect of the network structure in diffusion. However, this assumption is not valid for many diffusion types at both micro and macro scales. For instance, influenza spreads over human contact network, and this network is mostly heterogeneous. Similarly, influenza spreads between U.S. states at macroscale but the transmission rates between the states are not the same. Recently, some methods have been suggested to infer social networks from diffusion data. Among them, both NetInf [55] and MultiTree [117] formulate inference as a maximum likelihood problem in terms of only the edge existence, and ConNIe [102], NetRate [56], KernelCascade [38] and InfoPath [58] predict the edges by estimating the diffusion probabilities. Another network inference method makes a prior assumption about the scale-freenees of the network [31].

These methods have a number of shortcomings that we attempt to address here. They assume perfect knowledge of diffusion events, and neglect the possibility of partially observable, under-sampled probabilistic diffusion data. Further, they cannot model the uncertainty inherent in the diffusion data. Another shortcoming is their inability to estimate the diffusion rates at macroscale. In this case, existing methods cannot treat multiple nodes as a single ensemble node which is mandatory especially for large-scale networks. Lastly, we define the inference problem for arbitrary loss functions without making any prior assumption about the graph structure, and show that it can be solved optimally for certain type of loss functions.

Similar problems have been previously considered when collective statistics instead of individual statistics are available [41, 135]. For instance, collective graphical models are shown to be useful for estimating the bird migration paths given collective bird location data over time instead of individual positions [41] where they formulate inference as an extension of maximum flow problem. They also develop efficient approximate inference methods under more general collective graphical models [135]. However, these methods are based on flow conservation where latent nodes change position without changing their states over time by interacting with other latent nodes. Then, these methods cannot be directly applied to our problem of estimating the connectivity structure and influenza transmission rates at macroscale under SEIR.

4.2. Problem Formulation

Let $G = (V, E)$ be an unseen graph for which the edges E are difficult to observe directly. Edges of G may represent human contact events, interactions in PPI, relationships in social

Symbol	Definition
d	A single trace
D	Set of diffusion traces
T_d	Set of time points observed in D
$s_v^d(t_j)$, $e_v^d(t_j)$, $i_v^d(t_j)$, $r_v^d(t_j)$	Probabilities of v being in *S*, *E*, *I*, *R* states in trace d at time t_j
b	A perfect trace: $b = \{b_v,\ v \in V\}$
b_v	Perfect trace for node v: $b_v = \{t_{e,v}^b, t_{i,v}^b, t_{r,v}^b\}$
$t_{e,v}^b$, $t_{i,v}^b$, $t_{r,v}^b$	Exact time v passes into *E*, *I*, *R* in trace b

Table 4.1: Notation for problem definition

network, etc. We assume a uniform prior over the edges E since we do not have additional information about the graph structure, or the node attributes. At each time step, each node of G can be in one of several *states* $\mathcal{S}$. These states represent an abstraction of the node's status with respect to a diffusion process such as the spread of a virus. The model $\mathcal{M}$ governs how a node's state changes based on the states of its neighbors at previous time steps. Here, we focus on the general and widely-used SEIR model: the states $\mathcal{S}$ are Susceptible (*S*), Exposed but not contagious (*E*), Infected and contagious (*I*), and previously infected but now Recovered (*R*). The SI, SIS, SIR models are special cases of the SEIR model in which some states and transitions cannot occur. Those states are general enough abstractions to model various forms of diffusion in different contexts [91, 121]. The SEIR is Markovian, and it obeys the independent cascade [73] assumption which states that a single diffusion from one of node's neighbors is enough for node to become exposed.

More formally, a *trace* d of the SEIR diffusion process measured at time steps T_d provides us with a set of probabilities $\{s_v^d(t),\ e_v^d(t),\ i_v^d(t),\ r_v^d(t)\}$ for every node $v \in V$ and every time step $t \in T_d\}$, where $x_v^d(t)$ is the probability that node v is in state x at time t in trace d. For any node v and time t, we assume $s_v^d(t) + e_v^d(t) + i_v^d(t) + r_v^d(t) = 1$ indicating that v must be in one of the SEIR states. In fact, exact state transitions of node v into *E*, *I*, *R* states in trace d happen at $t_{e,v}^d$, $t_{i,v}^d$, $t_{r,v}^d$ respectively. We cannot observe these exact state transition times, but they are related to the observed trace d via the noise dynamics function $\mathcal{N}$ which is explained in detail in Section 4.4.1. $\mathcal{N}$ provide the probability of observing a particular probabilistic state trace for a node instead of the true state trace. Thus, our computational problem is:

Problem 1. *Infer the set of edges E given: the set of nodes V, a collection D of traces of probabilistic node states of the form described above, estimates of the noise dynamics $\mathcal{N}$, and a model $\mathcal{M}$, such as SEIR, by which the diffusion process is assumed to have occurred.*

Notation for the problem and its input is summarized in Table 4.1.

Our general framework for Problem 1 is this: we write down a set of probabilistic dynamic equations that model how the probability of each node being in each state changes under SEIR. This provides a theoretical trajectory through the space of state probabilities that depends on which edges exist in the graph and state transition times. We then formulate an optimization

Symbol	Definition
s_{uv}^d	Probability of diffusion from u to v in trace d
p_{uv}^d, f_{uv}^d	Probability, cumulative distribution of diffusion time from u to v in trace d
p_v^{ei}, p_v^{ir}	Probability distribution of $E \to I$, $I \to R$ transition time for v
$ei_v^d(t_j)$, $ir_v^d(t_j)$	Probability of $E \to I$, $I \to R$ transition for node v at time t_j
$\mathrm{p}_{uv}^d(t', t''\|t)$	Probability that v changed to *E* during $[t', t'']$ by u infected at t in trace d given u has not exposed v until t'
$ss_v^d(t_j)$	Probability that v does not leave state *S* between t_{j-1} and t_j
$\tilde{e}_v^d(t)$, $\tilde{i}_v^d(t)$, $\tilde{r}_v^d(t)$	Boolean indicator that is 1 if v enters *E*, *I*, *R* in trace d at time t
$g_s(a)$, $g_e(a)$, $g_i(a)$, $g_r(a)$	Probability of observing 4×1 state vector a instead of perfect *S*, ..., *R* states in any trace at any time.
α_m^s, α_m^e, α_m^i, α_m^r	Dirichlet distribution parameter vector for mixture component m and states *S*, ..., *R*

Table 4.2: Table of notation for diffusion model

problem to find the choice of edges that makes the theoretical trajectories match the observed traces as best as possible under the expectation of the selected loss function over the exact state transition times.

4.3. Diffusion Dynamics

We introduce x_{uv} for every pair of nodes $u \neq v$ with the interpretation that $x_{uv} = 1$ if edge (u, v) should exist. Assuming trace d is known for sorted time steps $T_d = t_1, t_2, t_3, \ldots, t_w$, for each consecutive pair t_{j-1}, t_j of this sample, SEIR can be thought as nonlinear discrete model and its dynamics can be written as in (4.1)–(4.4):

$$s_v^d(t_j) = s_v^d(t_{j-1})\, ss_v^d(t_j) \tag{4.1}$$

$$e_v^d(t_j) = e_v^d(t_{j-1})\, \left(1 - ei_v^d(t_j)\right) + s_v^d(t_{j-1})\, \left(1 - ss_v^d(t_j)\right) \tag{4.2}$$

$$i_v^d(t_j) = i_v^d(t_{j-1})\, \left(1 - ir_v^d(t_j)\right) + e_v^d(t_{j-1})\, ei_v^d(t_j) \tag{4.3}$$

$$r_v^d(t_j) = i_v^d(t_{j-1})\, ir_v^d(t_j) + r_v^d(t_{j-1}) \tag{4.4}$$

where $ss_v^d(t_j)$, $ei_v^d(t_j)$, and $ir_v^d(t_j)$ model the S→S, E→I, and I→R transition probabilities that will be explicitly defined ahead. The system of equations (4.1)–(4.4) give the probability of each node being in each state at time t_j. For instance, according to (4.2), node v is exposed at time t_j if it is exposed at t_{j-1} and has not transitioned into *infected* state, or it was susceptible at t_{j-1} and transitioned into *exposed* state. Among the all state transitions, only S→E is exogenous; it is affected by x_{uv} and that dependence is captured in $ss_v^d(t_j)$ terms. Figure 4.1 illustrates this dependence. Only the states of nodes 1, 4 may affect S→E transition for node v since there is an edge between v and them, while trace d_v provides the set of state probabilities of node v for a restricted set of time points.

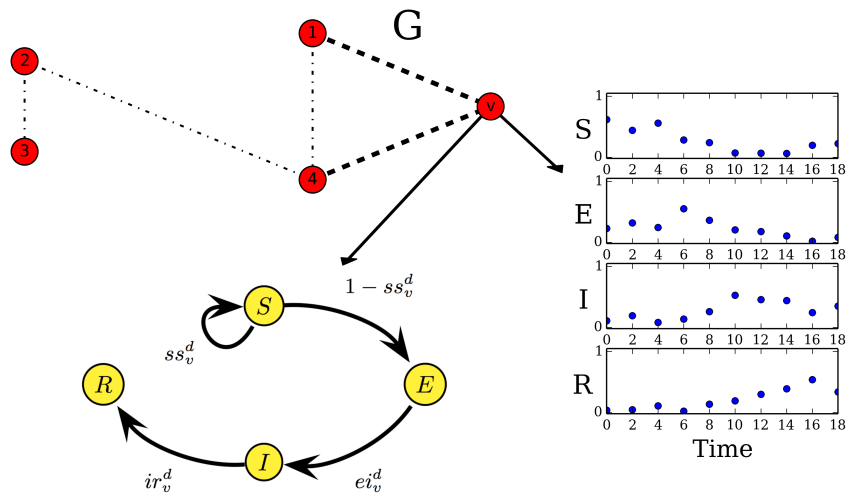


Figure 4.1: Only the S→E transition is being affected by G, while trace d_v provides the set of state probabilities of node v

In SEIR, v stays in state S at time t if it does not become exposed by an infected neighbor until after t. Let s_{uv}^d be the probability of diffusion from u to v in trace d. If d diffuses from u to v, diffusion from u to v happens at time $t_{i,u}^d + t$ where t is distributed according to given pmf p_{uv}^d, and its cdf f_{uv}^d. Then, probability that node u that was infected at time t would expose a neighbor v over interval $[t', t'']$ given it has not exposed v until t' ($\mathrm{p}_{uv}^d(t', t''|t)$) can be computed as in (4.5) by using Bayes rule when $t'' \geq t' \geq t$:

$$\begin{aligned}\mathrm{p}_{uv}^d(t', t''|t) &= \frac{P(u \text{ infected at } t \text{ exposed } v \text{ between } t' \text{ and } t'')}{P(u \text{ infected at } t \text{ has not exposed } v \text{ until } t')} \\ &= \frac{f_{uv}^d(t''-t)s_{uv}^d - f_{uv}^d(t'-t)s_{uv}^d}{1 - f_{uv}^d(t'-t)s_{uv}^d}\end{aligned} \tag{4.5}$$

where $f_{uv}^d(\Delta t)$ is the cdf of diffusion time from u to v in trace d, and the difference in the nominator is the probability of exposure from u in the interval $[t', t'']$. Using (4.5), we can estimate $ss_v^d(t_j)$ in (4.6) in terms of the probability of v not having been passed the infection from any node u:

$$ss_v^d(t_j) = \prod_{u \in V} \prod_{t < t_j} \left(1 - \mathrm{p}_{uv}^d(t_{j-1}, t_j|t)\right)^{x_{uv}\, \tilde{i}_u^d(t) \left(1 - \sum_{t' < t_j} \tilde{r}_u^d(t')\right)} \tag{4.6}$$

In other words, the probability that v remains susceptible at time t_j is estimated to be the product over all nodes u for which $x_{uv} = 1$ of the probability that u was infected at time $t < t_j$ without recovering until t_j but did not spread to v during the interval $[t_{j-1}, t_j]$. In (4.6), $\tilde{e}_v^d(t)$, $\tilde{i}_v^d(t)$ and $\tilde{r}_v^d(t)$ are boolean indicators that are 1 if v enters *E*, *I*, *R* in trace d at time t respectively ($t_{e,v}^d = t$, $t_{i,v}^d = t$, $t_{r,v}^d = t$). The probabilities $ei_v^d(t_j)$, $ir_v^d(t_j)$ in (4.1)–(4.4) can be estimated by (4.7)–(4.8)

in terms of E→I / I→R transition probabilities of v, and probability of v being E, I at time t.

$$ei_v^d(t_j) = \sum_{t=t_1}^{t_j} p_v^{ei}(t_j - t)\, \tilde{e}_v^d(t) \tag{4.7}$$

$$ir_v^d(t_j) = \sum_{t=t_1}^{t_j} p_v^{ir}(t_j - t)\, \tilde{i}_v^d(t) \tag{4.8}$$

A summary of the notation for the diffusion model is in Table 4.2.

4.4. Convex Risk (Expected Loss) Minimization Based Formulation

Having defined the diffusion dynamics, our goal is now to formulate the inference Problem 1. We assume that diffusion data is given, and we have an estimate of noise dynamics $\mathcal{N}$, so x_{uv} will be the only variables in diffusion dynamics (4.1)–(4.4). Let $b = \{b_v,\ v \in V\}$ be a noiseless trace, where $b_v = \{t^b_{e,v}, t^b_{i,v}, t^b_{r,v}\}$ and $t^b_{e,v}$, $t^b_{i,v}$, $t^b_{r,v}$ are the exact exposure, infection and recovery times of node v in perfect trace b respectively. Let B be a set of noiseless traces, and $L_b : X \times b \to R$, $L_B : X \times B \to R$ be real-valued loss functions that estimate the loss (cost) of the set of edges X given b and B respectively from the dynamic equations (4.1)–(4.4). In our case, set of true diffusion data B is hidden, but we observe D instead which defines the probabilities of being at states *S*, *E*, *I*, *R* for each time step as discussed in Section 4.2. Given D, the most probable set of edges $X \subseteq V \times V$ can be found by minimizing the risk (expected loss) over all realizations of D:

$$R(X, D) = \mathbb{E}_B[L_B] = \sum_B L_B(X, B)\, P(B|D) \tag{4.9}$$

where $P(B|D)$ models the noise dynamics $\mathcal{N}$; it is the probability that the set of observed traces D are generated from the latent true diffusion data B. We assume that each trace d is independent and noise affects each trace d independently, so $P(B|D) = \prod_{d \in D} P(b|d)$. Then, overall risk can be expressed as:

$$R(X, D) = \sum_{d \in D} R(X, d) = \sum_{d \in D} \sum_{b \in \mathcal{Q}(d)} L_b(X, b)\, P(b|d) \tag{4.10}$$

where $\mathcal{Q}(d) = \{(t_e(v), t_i(v), t_r(v)) \ :\ t_e(v) \in T_d,\ t_e(v) < t_i(v) < t_r(v)\,, v \in V\}$ is the set of all latent valid trace realizations that might explain the observed d.

4.4.1 Estimating $P(b|d)$

The noise affects each node independently, so $P(b|d) = \prod_{v \in V} P(b_v|d_v)$. From Bayes theorem, $P(b_v|d_v)$ can be expressed as:

$$P(b_v|d_v) = \frac{P(d_v|b_v)\ P(b_v)}{\underbrace{\sum_{b_v^* \in \mathcal{Q}(d)[v]} P(d_v|b_v^*)\ P(b_v^*)}_{P(d_v)}} \tag{4.11}$$

The probability $P(d_v|b_v)$ of observing d_v given b_v can be expressed as in (4.12) since observations at each time step are also independent:

$$P(d_v|b_v) = \prod_{t<t^b_{e,v}} g_s(d_v[t]) \prod_{t^b_{e,v} \le t < t^b_{i,v}} g_e(d_v[t]) \prod_{t^b_{i,v} \le t < t^b_{r,v}} g_i(d_v[t]) \prod_{t^b_{r,v} \le t} g_r(d_v[t]) \tag{4.12}$$

In (4.12), set of functions $g_x(d_v[t])$ for $x \in \{s, e, i, r\}$ give the probability of observing the 4×1 vector $d_v[t]$ at time t instead of perfect *S*, *E*, *I*, *R* traces respectively. Entries of $d_v[t]$ sum up to 1, so we model each $g_x(d_v[t])$ by a mixture of 4-dimensional Dirichlet distributions with M components as in (4.13) which may approximate any functional shape arbitrarily well:

$$g_x(d_v[t]) = \sum_{m \in M} w_m^x\, g_x^m(d_v[t]) \tag{4.13}$$

Each mixture component m for state x, trace d and time t is distributed according to the concentration parameters $\alpha_m^{x,d,t}$. For simplicity, we assume the same concentration parameters for every time t and trace d $\alpha_m^{x,d,t} = \alpha_m^x$. We also assume mixture weights w_m^x to be same for every trace d. Each Dirichlet component in (4.13) is explicitly written in (4.14) where $d_v^y[t]$ is the value of state y in $d_v[t]$, $\alpha_m^x[y]$ is the concentration parameter for state y, and $\mathbf{B}(\alpha_{\mathbf{m}}^{\mathbf{x}})$ is the normalizing constant:

$$g_x^m(d_v[t]) = \frac{1}{\mathbf{B}(\alpha_{\mathbf{m}}^{\mathbf{x}})} \prod_{y \in \{s,e,i,r\}} (d_v^y[t])^{\alpha_m^x[y]-1} \tag{4.14}$$

On the other hand, prior $P(b_v)$ in (4.11) can be explicitly written as in (4.15) in terms of state transition probabilities:

$$\begin{aligned} P(b_v) &= P(t^b_{e,v})\, P(t^b_{i,v}|t^b_{e,v})\, P(t^b_{r,v}|t^b_{i,v}) \\ P(b_v) &= P(t^b_{e,v})\, p_v^{ei}\, p_v^{ir} \end{aligned} \tag{4.15}$$

where $P(t^b_{i,v}|t^b_{e,v}) = p_v^{ei}$, $P(t^b_{r,v}|t^b_{i,v}) = p_v^{ir}$, and $P(t^b_{e,v}) = \frac{1}{|T_d|+1}$ is uniform since we do not have any prior information about the node transition times. The additional 1 in the denominator of $P(t^b_{e,v})$ models the case of v not ever becoming exposed.

The generative trace noise model expressed by (4.11) can also be seen as a variant of hidden semi-markov model (segment model) [101] where there is a hidden state for every time point in T_d with 4 possible values *S*, *E*, *I*, *R*. In our case, each state also emits a duration to model

the duration of being at a certain SEIR state, but each time step emits a distribution over 4 states instead of a single value as in basic hidden semi-markov model. Only a subset of state transitions are possible at each hidden state as they are restricted according to SEIR dynamics. Here, transition probabilities are defined by p_v^{ei}, p_v^{ir} and $P(t_{e,v}^b)$ whereas emission probabilities are from Dirichlet distribution mixture as in (4.13).

4.4.2 Estimating $L_b(X, b)$

There are variety of loss functions for $L_b(X, b)$. Here, we are dealing with the probabilities so we use negative log-likelihood loss ($L_b(X, b) = -\log(\mathcal{L}(X|b))$) where the likelihood is defined as in (4.16)–(4.17), and the risk function turns into (4.18).

$$\mathcal{L}(X|b) = \prod_{v\in V}\left(\prod_{t<t_{e,v}^b} s_v^d(t) \prod_{t_{e,v}^b\le t<t_{i,v}^b} e_v^d(t) \prod_{t_{i,v}^b\le t<t_{r,v}^b} i_v^d(t)\right) \tag{4.16}$$

$$\mathcal{L}(X|b) = C\prod_{v\in V}\left(\left(1-ss_v^d\left(t_{e,v}^b\right)\right) \prod_{t\in T_d, t<t_{e,v}^b} ss_v^d\left(t\right)\right) \tag{4.17}$$

$$R(X,D) = \sum_{d\in D}\sum_{b\in\mathcal{Q}(d)} P(b|d)\Bigg(\overbrace{\sum_{v\in V} -\log\left(1-ss_v^d\left(t_{e,v}^b\right)\right)}^{-\log(\mathcal{L}(X|b))} + \overbrace{\sum_{v\in V}\sum_{u\in V}\sum_{t_{i,u}^b\le t<\min(t_{r,u}^b, t_{e,v}^b)} -\log\left(1-\mathrm{p}_{uv}^d\left(t-1,t|t_{i,u}^b\right)\right)x_{uv}}^{}\Bigg) \tag{4.18}$$

Likelihood (4.16) is the multiplication of the node state probabilities at each observed time point in perfect trace b under SEIR. (4.17) is obtained from (4.16) by dynamic equations (4.1)–(4.4) where the constant C is obtained from the state transitions that do not involve X. Risk for negative log-likelihood loss is written explicitly in (4.18) when combined with (4.6), and it is convex as proven in Theorem 4.4.1. $R(X, D)$ (4.18) is convex so it can be minimized optimally by the existing convex optimization methods [16].

Theorem 4.4.1. *Risk $R(X, D)$ with negative log-likelihood loss function in* (4.18) *is convex.*

Proof. We need to prove the convexity of the each additive term w.r.t. X to prove the convexity of $R(X, D)$ (4.18) for negative log-likelihood loss. There are two types of terms involving X: $-\log\left(1-\mathrm{p}_{uv}^d(t_{e,u}^b, t-1, t)\right)x_{uv}$ and $-\log\left(1-ss_v^d\left(t_{e,v}^b\right)\right)$. $-\log\left(1-\mathrm{p}_{uv}^d(t_{e,u}^b, t-1, t)\right)x_{uv}$ is convex since it is a linear function of X. The other term $-\log\left(1-ss_v^d\left(t_{e,v}^b\right)\right)$ can be explicitly written in (4.19):

$$-\log\left(1-ss_v^d\left(t_{e,v}^b\right)\right) = -\log\left(1-\exp^{\sum_{u\in V,\, t_{i,u}^b<t_{e,v}^b} w_u\, x_{uv}}\right) \tag{4.19}$$

where w_u are defined in (4.20) for every $u \in V$, $t_{i,u}^b < t_{e,v}^b$:

$$w_u = \log\left(1 - \mathrm{p}^d_{uv}(t^b_{i,u}, t^b_{e,v} - 1, t^b_{e,v})\right) \tag{4.20}$$

(4.19) is convex since its Hessian when expressed in (4.21):

$$H = \frac{\exp^{\sum_{u\in V,\, t^b_{i,u}<t^b_{e,v}} w_u\, x_{uv}}}{\left(1 - \exp^{\sum_{u\in V,\, t^b_{i,u}<t^b_{e,v}} w_u\, x_{uv}}\right)^2} \begin{bmatrix} w_1^2 & w_1w_2 & w_1w_3 & \dots & w_1w_n \\ w_2w_1 & w_2^2 & w_2w_3 & \dots & w_2w_n \\ w_3w_1 & w_3w_2 & w_3^2 & \dots & w_3w_n \\ \dots & \dots & \dots & \dots & \dots \\ w_nw_1 & w_nw_2 & w_nw_3 & \dots & w_n^2 \end{bmatrix} \tag{4.21}$$

is Positive semidefinite (PSD) as it can be expressed as $Z\, y^T\, y$ where:

$$y = [\, w_1,\, w_2,\, \dots,\, w_n] \tag{4.22}$$

$$Z = \frac{\exp^{\sum_{u\in V,\, t^b_{i,u}<t^b_{e,v}} w_u\, x_{uv}}}{\left(1 - \exp^{\sum_{u\in V,\, t^b_{i,u}<t^b_{e,v}} w_u\, x_{uv}}\right)^2} \geq 0 \tag{4.23}$$

□

4.4.3 A More Efficient Relaxation

However, minimizing $R(X, D)$ (4.18) requires estimating the expectation of the loss function over set of all possible perfect transition time realizations defined by $\mathcal{Q}(d)$. This expectation estimation can be quite time-consuming since it may require an exponential number of summations in the worst case. To infer graphs efficiently, we can instead optimize the relaxed risk ($\hat{\mathcal{R}}(X, D)$) as in (4.24):

$$\hat{\mathcal{R}}(X, D) = \sum_{d\in D} \sum_{b\in\mathcal{Q}(d)} P(b|d) \left(\sum_{v\in V} \mathcal{T}^b_v + \sum_{v\in V} \sum_{u\in V} \sum_{t=t^b_{i,u}}^{\min(t^b_{r,u}, t^b_{e,v})} - \log\left(1 - \mathrm{p}^d_{uv}(t-1, t|t^b_{i,u})\right) x_{uv} \right) \tag{4.24}$$

which is obtained by replacing each nonlinear term $\log(1 - ss^d_v(t_j))$ with its first-order Taylor approximation ($\mathcal{T}^b_v$) as estimated in (4.25):

$$\mathcal{T}^b_v = \sum_{u\in V} \log\left(\mathrm{p}^d_{uv}(t^b_{e,v} - 1, t^b_{e,v}|t^b_{i,u})\right)\ (x_{uv} - 1) \tag{4.25}$$

We have $P(b|d) = \prod_{v\in V}\ P(b_v|d_v)$ due to independence of noise for every node, so (4.24) becomes:

$$\hat{\mathcal{R}}(X, D) = \sum_{d\in D} \sum_{b\in\mathcal{Q}(d)} \sum_{u,v\in V\times V} P(b_u|d_u)\, P(b_v|d_v)\, \mathbf{M^b_{uv}}\, x_{uv}\ +\ C \tag{4.26}$$

where

$$\mathbf{M^b_{uv}} = \log(\mathrm{p}^d_{uv}(t^b_{e,v} - 1, t^b_{e,v}|t^b_{i,u})) - \sum_{t^b_{i,u} \le t < \min(t^b_{r,u}, t^b_{e,v})} \log(1 - \mathrm{p}^d_{uv}(t-1, t|t^b_{e,u})) \tag{4.27}$$

Equation (4.26) is a linear function of X. In (4.26), each x_{uv} depends only on the exact state transition times of u and v since the rest of the probabilities in $P(b|d)$ marginalize out when written as $P(b|d) = \prod_{v \in V} P(b_v|d_v)$.

We can express linear Eqn. (4.26) more explicitly in tensor form by (4.28) since expected loss for each edge (u, v) depends only on the exact exposure time from $P_v(b|d)$ (sender), and exact infection and recovery times from $P_u(b|d)$ (receiver).

$$\hat{\mathcal{R}}(X, D) = \sum_{d \in D} \sum_{v \in V} \sum_{u \in V} \sum_{t^i_u \in T_d} \sum_{t^i_u < t^e_v} \sum_{t^e_v \le t^r_u} \left(\mathbf{P^d_{v,e}}\left[t^e_v\right] \times \mathbf{P^d_{u,i,r}}\left[t^i_u, t^r_u\right] \mathbf{M^d_{uv}}\left[t^i_u, t^r_u, t^e_v\right] \, x_{uv} \right) \tag{4.28}$$

In (4.28), $(|T_d|+1)\times 1$ vector $\mathbf{P^d_{v,e}}\left[t^e_v\right]$, and $(|T_d|+1)\times(|T_d|+1)$ matrix $\mathbf{P^d_{u,i,r}}\left[t^i_u, t^r_u\right]$ express these marginal probability distributions as defined in (4.29)–(4.30). In both equations, the $(|T_d|+1)$'th entries model the case of never transitioning into the corresponding state:

$$\mathbf{P^d_{v,e}}\left[t^e_v\right] = \begin{cases} \sum_{t^e_v < t_1} \sum_{t_1 < t_2} P_v(b = \{t^e_v, t_1, t_2\}|d) & \text{if } t \in T_d \\ 1 - \sum_{t \in T_d} \mathbf{P^d_{v,e}}\left[t\right] & \text{else} \end{cases} \tag{4.29}$$

$$\mathbf{P^d_{u,i,r}}\left[t^i_u, t^r_u\right] = \begin{cases} \sum_{t_1 < t^i_u} P_v(b = \{t_1, t^i_v, t^r_v\}|d) & \text{if } t^i_u < t^r_u \\ 0 & \text{else} \end{cases} \tag{4.30}$$

The $(|T_d|+1)^3$ size tensor $\mathbf{M^d_{uv}}\left[t^i_u, t^r_u, t^e_v\right]$ in (4.28) defines the coefficients for the edge from u to v to exist under the transition times $t^i_u,\ t^r_u,\ t^e_v \ \in \ (T_d + 1)^3$ as explicitly defined below in (4.31):

$$\mathbf{M^d_{uv}}\left[t^i_u, t^r_u, t^e_v\right] = \begin{cases} \log\left(\mathrm{p}^d_{uv}\left(t^e_v - 1, t^e_v|t^i_u\right)\right) \text{ if } t^i_u < t^e_v \le t^r_u \\ -\sum_{t < t^e_v} \log\left(1 - \mathrm{p}^d_{uv}\left(t-1, t|t^e_u\right)\right) \\ 0 \qquad\qquad\qquad\qquad \text{else} \end{cases} \tag{4.31}$$

We can express (4.28) more compactly by (4.32) where each x_{uv} coefficient is inner product of third-order tensor $\mathbf{M^d_{uv}}$ and the vector $\mathbf{P^d_{v,e}}$, and then it is sum of the entries of Hadamard product of the resulting matrix and matrix $\mathbf{P^d_{u,i,r}}$:

$$\hat{\mathcal{R}}(X, D) = \sum_{d \in D} \sum_{v \in V} \sum_{u \in V} \sum_{jk} \left(\mathbf{P^d_{u,i,r}} \odot \left(\mathbf{M^d_{uv}} \cdot \mathbf{P^d_{v,e}} \right) \right) \, x_{uv} \tag{4.32}$$

$\hat{\mathcal{R}}(X, D)$ (4.32) is linear, convex, and it can be optimized quite fast since we can estimate all x_{uv} coefficients by $O(|D||V|^2 \max(|T_d|)^3)$ operations instead of $O(|D||V|^2 \max(|T_d|)^V)$. X can be found by minimizing $\hat{\mathcal{R}}(X, D)$ optimally by Program (4.33)–(4.35):

$$\underset{X}{\operatorname{argmin}} \quad \hat{\mathcal{R}}(X, D) + \lambda \sum_{(u,v)\in V\times V} x_{uv} \tag{4.33}$$

$$\text{s.t.} \quad \sum_{u\in V,\, t_u^i < t_v^e \le t_u^r} x_{uv} \ge 1,\ \forall d \in D, v \in V \tag{4.34}$$

$$0 \le x_{uv} \le 1, \quad \forall (u, v) \in V \times V \tag{4.35}$$

Covering constraints (4.34) make sure that at least single edge exists between the newly infected node v and the previously infected nodes that are not yet recovered for every trace d. When the diffusion data is not perfect, (4.34) are removed since we do not know $t_{i,u}^d$, $t_{e,v}^d$, $t_{r,u}^d$ from given diffusion data. We obtain the binary solution by randomly rounding x_{uv}.

4.5. Possible Improvements

4.5.1 Estimating Noise Dynamics Simultaneously With Graph Inference

We may not always know the noise dynamics parameters in Problem 1. In this case, we minimize the expected loss to simultaneously estimate the most possible X and noise parameters under the generative noise model described in Section 4.4.1. However, their joint optimization is not convex anymore even for negative log-likelihood loss function.

To efficiently estimate both, we propose a two-step procedure similar to Monte Carlo Expectation Maximization [4]. In the first step, we estimate the optimal set of edges X given D and the estimated noise dynamics parameters, and we estimate the optimal noise dynamics parameters given X and D in the second step. First step is same as solving Problem 1, and both steps alternate until convergence to a local optimum. In the second step, we try to find the best mixture weights w_m^x assuming dirichlet distribution concentration parameters α_m^x are fixed at uniformly sampled locations on a four-dimensional grid. Optimizing for the best w_m^x over all latent valid trace realizations B quickly becomes intractable for large number of traces, so we sample set of latent traces by turning the log-likelihoods estimated in the first step into probabilities via exponentiation. Let $\overline{B}$ be the set of sampled latent traces, and $W = \{w_{mx} \,|\, m \in M,\ x \in s, e, i, r\}$ be the set of weight variables where w_{mx} is weight of mixture component m for state x. Given $\overline{B}$, we minimize the negative logarithm of multiplications of the probabilities for $\overline{B}$ as in:

$$\mathcal{L}^p(W|X, D) = \sum_{b\in\overline{B}} \sum_{v\in V} \sum_{t\in T^b} -\log\left(\sum_{m\in M} w_{my} e_m^{bvt}\right) \tag{4.36}$$

where y is the state of node v at time t in trace b, and e_m^{bvt} are the coefficients estimated over fixed α_m^x's by Equation (4.14). Then, we solve the following Program (4.37)–(4.39) to estimate W:

$$\underset{W}{\operatorname{argmin}} \quad \mathcal{L}^p(W|X,D) \tag{4.37}$$

$$\text{s.t.} \sum_{m\in M} w_{mx} = 1, \quad \forall x \in s,e,i,r \tag{4.38}$$

$$w_{mx} \geq 0, \quad \forall m \in M, \forall x \in s,e,i,r \tag{4.39}$$

This optimization program is not under-constrained since we assume same mixture weights for each node which makes a total of $4M$ variables. Objective (4.37) is convex as in Theorem 4.5.1 which proof follows from the fact that convexity is preserved under addition and negative logarithm of weighted multivariate linear function is also convex due to its positive semidefinite hessian matrix.

Theorem 4.5.1. *Objective* (4.37) *is convex.*

Program (4.37)–(4.39) can be solved optimally by exponentiated gradient descent algorithm [75] since equality constraints (4.38) are non-overlapping. In this case, exponentiated gradient updates involve:

$$w_{mx}^{t+1} = \frac{w_{mx}^t \exp^{\left(-\eta \nabla_{mx}(w_{mx}^t)\right)}}{Z_x^t} \tag{4.40}$$

where $Z_x^t = \sum_{m\in M} w_{mx}^t \exp^{\left(-\eta \nabla_{mx}(w_{mx}^t)\right)}$ is the state-dependent normalization constant, parameter $\eta > 0$ is the learning rate, and $\nabla_{mx}(w_{mx}^t)$ is the gradient of objective (4.37) with respect to w_{mx}. Weights estimated by (4.40) already satisfy the constraints (4.38), and this method iterates until convergence.

4.5.2 Improvements For Special Cases of SEIR

Most of the expressions in the previous sections become slightly easier for SI and SIR models due to fewer states, and disappearance and modifications of the certain transitions. For instance, (4.15) turns into the uniform distribution for SI model since p_v^{ei}, p_v^{ir} transitions disappear, and we do not have any prior information about the infection times. We estimate the coefficients of the relaxed risk $\hat{\mathcal{R}}(X,D)$ by $\mathbf{M}_{\mathbf{uv}}^{\mathbf{d}}[t_u^i, t_v^i, t_u^r]$, $\mathbf{P}_{\mathbf{v,i}}^{\mathbf{d}}$ and $\mathbf{P}_{\mathbf{u,i,r}}^{\mathbf{d}}$ for SIR model. However, $\mathbf{M}_{\mathbf{uv}}^{\mathbf{d}}[t_u^i, t_v^i]$ becomes a second-order tensor (matrix) for SI due to the disappearance of recovery times, and we use it together with the vectors $\mathbf{P}_{\mathbf{v,i}}^{\mathbf{d}}$ and $\mathbf{P}_{\mathbf{u,i}}^{\mathbf{d}}$ to estimate $\hat{\mathcal{R}}(X,D)$ coefficients by $O(|D||V|^2 \max(|T_d|)^2)$ operations.

4.5.3 *CORMIN* Speedups

We speed-up *CORMIN* substantially via two improvements: Edge inference for each node is independent of each other, so risk minimization problem for each node can be solved optimally in parallel which makes *CORMIN* scalable to large graphs as in [56]. Secondly, when estimating the tensor multiplication in (4.32) for traces with large T_d, we approximate the resulting coefficients by building the tensors (4.29)–(4.30) and (4.31) for subset of time points by sampling them via MCMC. The coefficients estimated by ignoring subset of time points are good approximations,

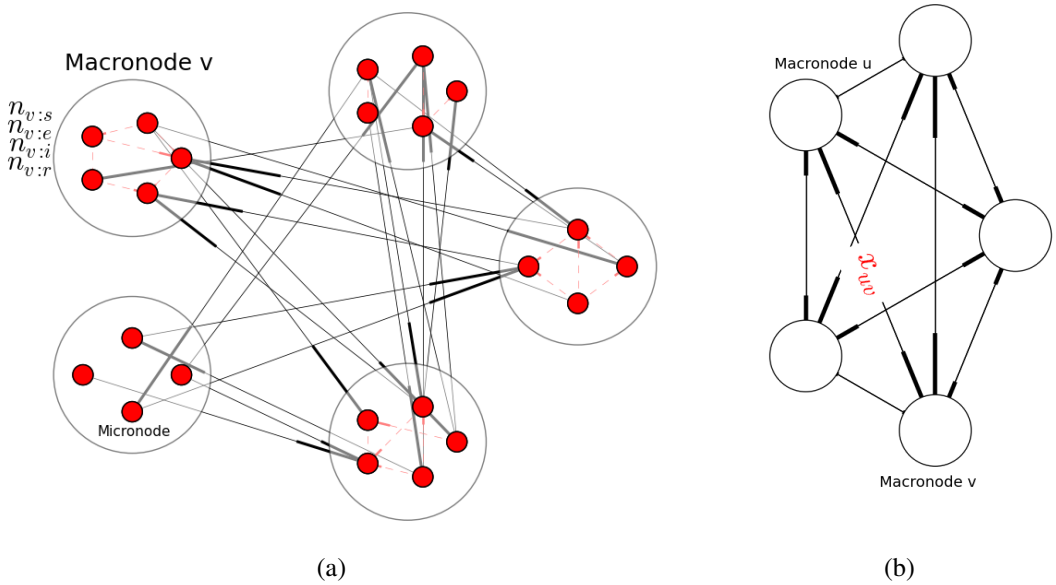
as well as *CORMIN* can infer graphs reasonably well in several minutes from the traces that are sampled at a high rate.

4.5.4 Caveats

In Problem (1), we assume DM parameters between consecutive time steps to be independent and uncorrelated. However, noise dynamics in many realistic scenarios can be better modeled by time-sensitive Dirichlet Mixture model where Dirichlet mixture parameters are also correlated across different time points. These additional dependence constraints further reduce the solution space. We leave improving *CORMIN* to handle such caveats as a future work.

4.6. Macroscale Inference

In Section (4.4), we focused on inferring the exact connectivity structure which may be a human-contact network at high school or Facebook friendship network. However, we may not be always interested in inferring the exact network structure since (1) networks we are considering may be massively large, and available diffusion data may not be enough for large-scale inference over them, and (2) it is not worth inferring the every single edge as the connectivity structure at a higher level may be enough for our purpose. For instance, it is impossible to infer the whole human contact network or influenza diffusion network in U.S. from the available influenza diffusion data. Additionally, understanding the U.S. influenza network at the macroscale, such as inferring the diffusion rates between U.S. states rather than between the humans, may be enough to take preventive measures to stop epidemics.



(a) (b)

Figure 4.2: a) The original network, b) The same network at macroscale from our perspective

At macroscale connectivity level, each macronode is composed of micronodes as in Figure (4.2a), and we are rather interested in estimating the ensemble connectivity rates between and inside the macronodes instead of between the single micronodes as in Figure (4.2b). More formally, we define a fully connected weighted graph $G = (V_m, E_m)$ with self-loops where V_m are macronodes, every edge (u, v) in E_m has an associated macro connectivity rate x_{uv} that the two random micronodes between u and v are connected, and self-loops model the diffusion inside each macronode. Additionally, we assume that the connectivity inside and between every macronode pair is *homogenous* which is quite realistic for large uniform macronodes.

Let n_v^t be the number of micronodes inside macronode v at time t, and $n_{v:s}^t$, $n_{v:e}^t$, $n_{v:i}^t$, $n_{v:r}^t$ be the number of micronodes inside macronode v belonging to *S*, *E*, *I*, *R* states respectively. Similarly, we define $p_{v:s}^t = \frac{n_{v:s}^t}{n_v^t}$, $p_{v:e}^t = \frac{n_{v:e}^t}{n_v^t}$, $p_{v:i}^t = \frac{n_{v:i}^t}{n_v^t}$ and $p_{v:r}^t = \frac{n_{v:r}^t}{n_v^t}$ as the fractions of micronodes in the corresponding SEIR states at time t, and let $\hat{p}_{v:i}^t = p_{v:i}^t - p_{v:i}^{t-1}$, $\hat{p}_{v:e}^t$, $\hat{p}_{v:r}^t$ be the fraction of newly infected, exposed, recovered nodes respectively. In this case, set of $\hat{p}_{v:x}^t$ for each macronode $v \in V$, each state $x \in \{s, e, i, r\}$, and each time step t define the diffusion data for Problem (1) at macroscale where we do not know exactly which micronode got infected or recovered. This diffusion data has a natural interpretation: each $\hat{p}_{v:i}^t$ is the probability that a random micronode in v has transitioned to state *I*. At this scale, we estimate the ensemble connectivity rates x_{uv} by optimally minimizing the non-relaxed version of the objective (4.18) where negative log-likelihood is modified as follows:

$$
\begin{aligned}
-\log\left(\mathcal{L}(X|b)\right) = \sum_{v \in V} -n_{v:e}^{t_{e,v}^b} \log\left(1 - ss_v^d\left(t_{e,v}^b\right)\right) + \sum_{v \in V} \sum_{u \in V} \\
\sum_{t_{i,u}^b \le t < \min(t_{r,u}^b, t_{e,v}^b)} -n_{v:s}^t n_{u:i}^{t_{i,u}^b} \log\left(1 - \mathrm{p}_{uv}^d\left(t-1, t | t_{i,u}^b\right)\right) x_{uv}
\end{aligned}
\tag{4.41}
$$

where log probabilities of each macronode are also multiplied by the number of micronodes since micronodes are independent and multiplications become summation by taking the logarithm. Solution is then obtained without rounding X.

4.7. Experiments & Results

4.7.1 Synthetic Networks and Trace Generation

We tested the inference performance over synthetic networks as follows: We generated 10 synthetic networks of 500 nodes and 5000 edges from each of DMC [148], LPA [10], ForestFire (FF) [90] and Erdos-Renyi (RDS) models by sampling uniformly through their parameters space. Each synthetic trace was generated by choosing a source node randomly and running the diffusion over the network until either all nodes become recovered (or infected under the SI model) or until the spread dies out. When a node gets infection from multiple nodes at different times, it is infected at the earliest infection time. Given noise ratio p between 0 and 1, we added synthetic noise as follows: For every node and time step, we assign probabilistic state vector sample obtained from Dirichlet distribution with concentration parameter vector $\alpha = \left[\frac{p}{4}, \frac{p}{4}, \frac{p}{4}, 1 - \frac{3p}{4}\right]$

where $1 - \frac{3p}{4}$ is the concentration parameter for the current state. This parameter vector becomes uniform for higher noise levels, where it becomes almost impossible to recover the original state.

4.7.2 Real Networks

We tested *CORMIN* by modeling influenza spreading over the human contact network, called *Contact-static*, at an American high school [121] as SI, SIR, SEIR. In this network, nodes represent people and an edge exists between two people if they are near each other. We simulated influenza spreading with $s_{uv} = 0.2$, p_{uv} as weibull distribution with ($\lambda = 9.5, k = 2.3$), and p_v^{ei}, p_v^{ir} as exponential distributions with $\lambda = 0.5$ and $\lambda = 0.2$ respectively as discussed in [153]. We also inferred the average influenza transmission rates between U.S. states at macroscale by using the Google Flu Trends Data between 2003–2013 treating each influenza season from September through May as an independent trace where each week is modeled by a single time step. In this *Macro-state* network, each node represents a U.S. state, edges model the influenza transmission rates between those states, and the graph has self-loops to model the influenza diffusion inside the states. The probability of infection at each time step at each U.S. state is the percentage of the people affected by influenza in that state in the corresponding week.

4.7.3 Experiment Details

We implemented *CORMIN* using CPLEX. Its code, used datasets are available on the web[1]. Edge inference for each node is independent and can be solved optimally in parallel which makes *CORMIN* scalable to large graphs. *CORMIN* is reasonably fast; it can infer a graph of 500 nodes and 5000 edges from 100 traces in less than a minute on personal laptop. We compared the performance of *CORMIN* with the best-performing existing methods MultiTree [117], NetRate [56], NetInf [55] and InfoPath [58]. We run MultiTree and NetInf giving them the exact number of edges in the true graph although such a perfect estimate is not available a priori. When the diffusion data is perfect, we run *CORMIN* nonparametricly by using only the covering constraints, and estimate the sparsity parameter λ in (4.33) by cross-validation when the diffusion data is partially observable. In this case, we performed 5 cross-validation over the diffusion data as follows: We estimate the set of edges from the training part of the diffusion data for 500 λ parameters between 0 and 100, and estimate the error of observing the remaining traces over the inferred graph for every λ. After repeating this for 5 parts, we return the λ minimizing the total error.

When estimating the prediction score at microscale, the edges of the unknown graphs are the positive examples and the pairs between which no edge exists are the negatives. Unknown graphs are sparse so we measure the performance by both $F1 = \frac{2\,\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$ and $F0.1 = \frac{1.01\,\text{precision}\times\text{recall}}{0.01\,\text{precision}+\text{recall}}$ to put more weight on precision where precision is the fraction of edges in the inferred network that are also present in the true network, and recall is the fraction of edges in the true network that are also present in the inferred network. We evaluated the performance of *CORMIN* at macroscale by estimating Pearson correlation coefficient between the

[1] `http://www.cs.cmu.edu/~ckingsf/software/cormin/`

inferred influenza rates and the transportation rates between U.S. states estimated from Gowalla dataset [26].

4.7.4 Inferring a Static Human Contact Network

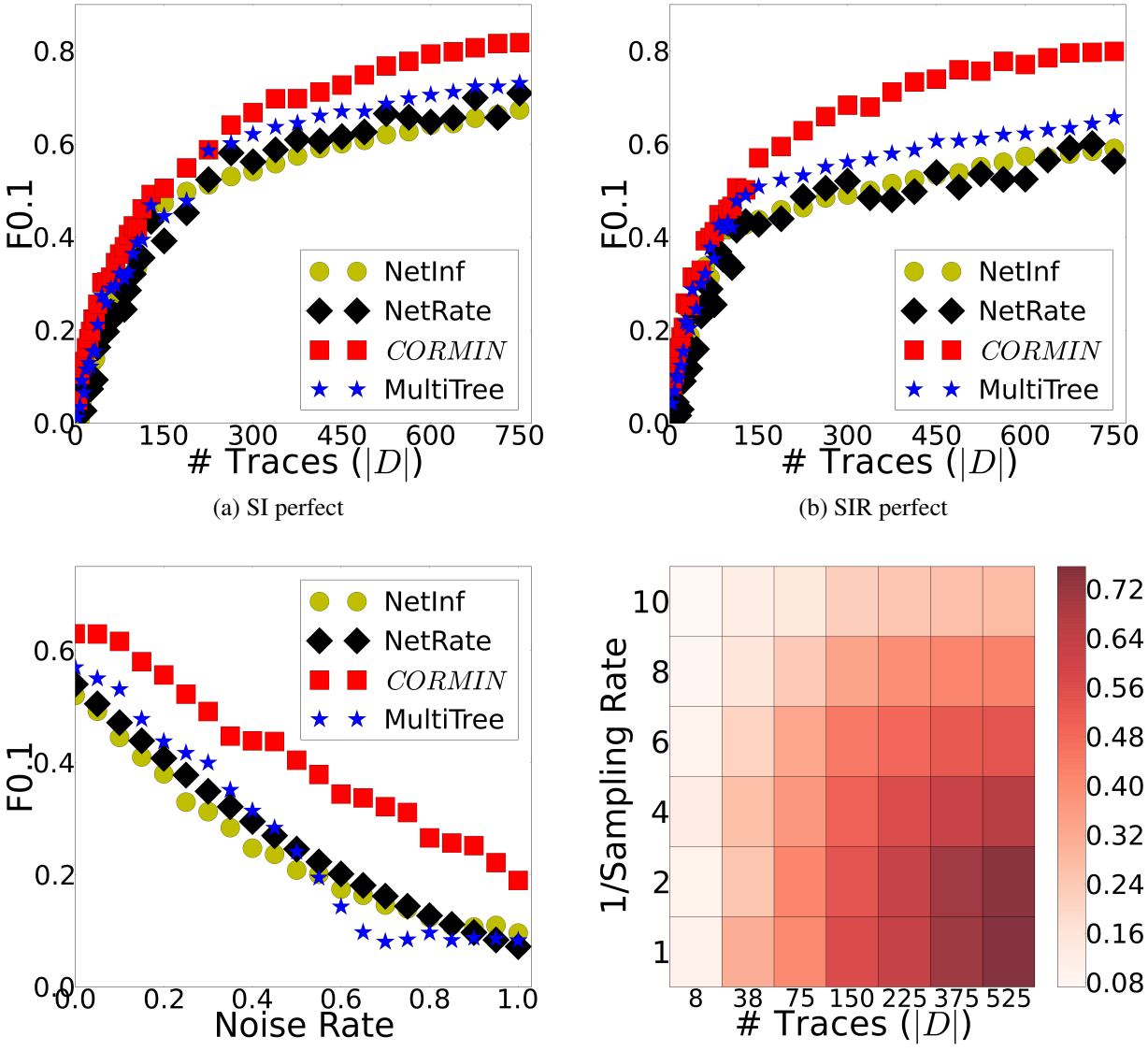
We inferred *Contact-static* by synthetic influenza traces on SI, SIR, SEIR that are generated from the real influenza diffusion parameters as discussed above. In these influenza traces, the infected state models the human infected with the influenza that is also spreading it to the other people, whereas the exposed state models the human infected with the influenza but has not yet started spreading it to the rest of the school network. When the diffusion data is perfect, *CORMIN* performs the best even though it is nonparametric as in Figure 4.3a. Similarly, *CORMIN* performs the best under SIR as in Figure 4.3b, and the performance difference between *CORMIN* and the existing methods are greater than in Figure 4.3a.

The performance difference between *CORMIN* with the sparsity parameter λ estimated by cross-validation and the existing methods becomes more significant when the diffusion data is noisy. This noisy data case is realistic: it may be too costly to track the influenza dynamics exactly since influenza symptoms may be confused with other symptoms, and the diffusion data may be limited especially for novel influenza types such as H5N1 [97] when they first appeared. According to Figure 4.3a, *CORMIN* achieves $F0.1$ score of 0.7 from 350 perfect traces, and it can achieve the same score from approximately 700 noisy traces. In contrast to this performance, the existing methods can only achieve $F0.1$ score of 0.5 from the same noisy traces. When plotted against increasing noise levels as in Figure 4.3c, *CORMIN* can achieve $F0.1$ score greater than 0.4 even from highly corrupted traces whereas the existing methods are significantly affected by the increasing noise levels, as $F0.1$ for all of them quickly drop below 0.2.

Diffusion data sampled at a lower rate provides less information, and this leads to a overall decrease in *CORMIN*'s performance as in Figure 4.3d where $\frac{1}{x}$ rate means we only observe 1 time point in every x-length interval. *CORMIN*'s performance is affected by the lower sampling rates, but its performance is still reasonable for sampling rates higher than $\frac{1}{5}$ for SEIR across various numbers of diffusion traces. In summary, *CORMIN* performs well on both perfect and partially observable data, and its performance is less affected by the noise in the diffusion data which is not the case for the existing methods.

CORMIN can reasonably reveal the human contacts as in Figure 4.4 which shows the random 50 node subgraph of both estimated and the true contact networks from 50 and 200 diffusion traces respectively. In Figure 4.4, gray edges represent the edges that are correctly predicted by *CORMIN*, red edges represent the edges that are in the true contact network but not in the estimated network, and the blue edges represent the edges that are in the estimated network but not in the true contact network. In terms of nodes, red nodes represent the students, green and black nodes represent the teachers and the school staff respectively. According to Figure 4.4, students are densely connected with each other, and the most of the mispredicted connections are between the students instead of between the rest of the people.

Networks inferred by *CORMIN* closely mimick the full range of properties of the true network even from a limited number of traces. Comparison of some of the metrics of *Contact-static* estimated from 50 traces, and the true *Contact-static* can be seen in Table 4.3. For instance, we know that human contact network has scale-free degree distribution with exponent 2.254, and



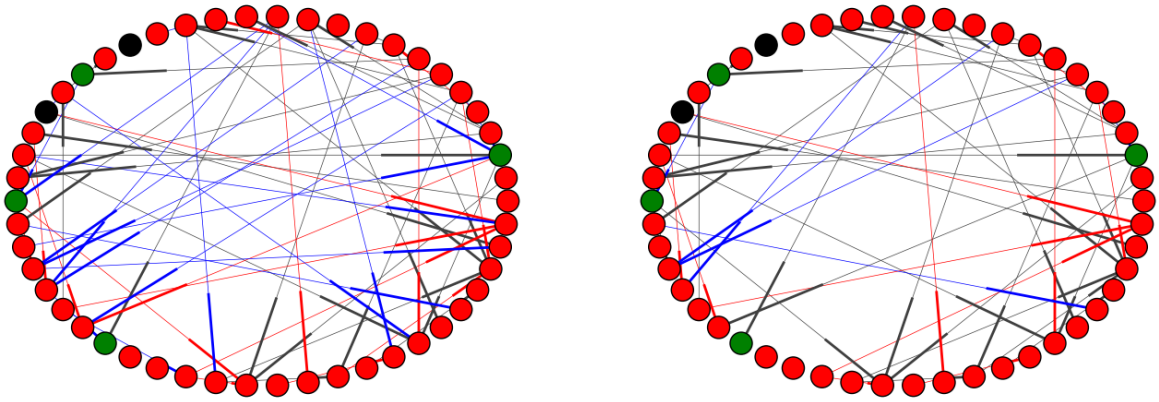
(a) SI perfect

(b) SIR perfect

(c) SI partial

(d) SEIR Heatmap

Figure 4.3: $F0.1$ vs. number of traces for *Contact-static* under (a) SI, (b) SIR from perfect data; c) $F0.1$ vs. noise ratio for *Contact-static* under SI from 250 traces, d) $F0.1$ Heatmap of number of traces vs. sampling rate under SEIR



a) 50 traces b) 200 traces

Figure 4.4: 50 node subgraph of true and the estimated *Contact-static* from *CORMIN* under SI from a) 50 traces, b) 200 traces

the network estimated by *CORMIN* has similar exponent 2.072.

	Estimated	Truth
Modularity [110]	0.67	0.73
Scale-free exponent	2.072	2.254
Assortativity	0.141	0.121
Avg. Clustering Coefficient	0.23	0.261
Diameter	10	8

Table 4.3: Metrics of true and estimated *Contact-static* networks from 50 traces

4.7.5 Estimating Influenza Diffusion Rates Between U.S. States

We estimated the average influenza diffusion rates between U.S. states at macroscale from Google Flu Trends data as described in Section 4.7.2 without rounding the resulting x_{uv}. Google Flu Trends data shows the number of weekly infections at each U.S. state between 2003-2013. Here, each node in the network represents a U.S. state, and we treat each influenza season from September to May as an independent trace. Google Flu Trends data is incomplete so we completed the missing data for states at each week as the average of the neighbouring states.

True diffusion rates are unknown but we compared the inferred influenza rates with the transportation rates estimated from Gowalla dataset [26]. Estimated ensemble diffusion rates between the most populated 16 U.S. states are shown in Figure 4.5. Diagonal entries are the diffusion rates inside U.S. states, and we found influenza diffusion rates inside the most populated states such

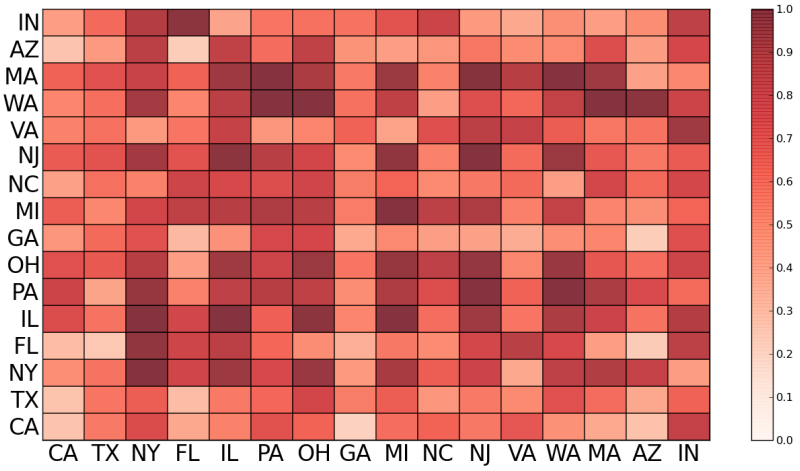


Figure 4.5: Estimated influenza transmission rates between the most populated 16 U.S. states over *Macro-state* network

as New York, Illinois and Texas to be the highest as well as between the nearby states. We estimated the diffusion rates between the northern states to be higher than the diffusion rates for the southern states. However, one may approach these results with caution since the diffusion rates estimated over Google Flu Trends data may be a slight overestimate as discussed in [86].

We estimated the Pearson correlation coefficient between the estimated influenza diffusion rates and transportation rates to be 0.32 which shows that the transportation is one of the major contributors in influenza transmission between U.S. states as discussed previously [8]. We also found influenza diffusion between U.S. states to be fairly asymmetric where we define the asymmetry of the rate matrix as the average of the absolute differences between diffusion rates of every pair of entries the symmetric entries, and estimated it as 0.15 for our rate matrix.

To quantify the degrees of importance of U.S. states in influenza diffusion, we estimated the hubs and authorities values (HITS) for U.S. states on the inferred network by [77]. In general, a good hub represents a U.S. state that diffuses influenza to many other U.S. states, and a good authority represents a U.S. state that gets influenza from other states without much spreading it to the other states. Table 4.4 shows the hubs and authorities scores of some U.S. states.

	Hubs	Authorities		Hubs	Authorities
California	0.058	0.060	Illinois	0.068	0.071
Texas	0.052	0.052	Pennsylvania	0.073	0.071
Michigan	0.066	0.070	Massachusetts	0.071	0.068
Florida	0.060	0.056	Washington	0.074	0.068

Table 4.4: Hubs and authorities scores of some U.S. states on *CORMIN* estimated macroscale network

In general, almost all states tend to have close hub and authority scores. We found some of

	FF			LPA			DMC			RDS		
	SI	SIR	SEIR	SI	SIR	SEIR	SI	SIR	SEIR	SI	SIR	SEIR
CORMIN	**0.62**	**0.57**	**0.61**	**0.59**	**0.5**	**0.61**	0.45	0.44	0.49	0.52	**0.53**	**0.55**
MultiTree	0.54	0.47	0.46	0.51	0.43	0.45	0.44	0.34	0.45	0.49	0.35	0.4
NetInf	0.52	0.45	0.46	0.50	0.42	0.47	0.4	0.41	0.47	0.47	0.33	0.38
NetRate	0.45	0.5	0.43	0.52	0.42	0.44	0.41	0.39	0.47	0.45	0.28	0.36

Table 4.5: $F1$ vs. growth and diffusion models for synthetic graphs inferred using 250 traces (No noise added)

the northern states such as Washington and Massachusetts as well as some mid U.S. states such as Virginia to have higher hub scores whereas the most of the southern states either have slightly higher authority scores or they have close hub and authority scores. Overall, we may think the top-scoring hubs as diffusion accelerators whereas the top-scoring authorities slow down the epidemics. Depending on whether a state is a hub or an authority, we may take different types of measures to prevent or slowdown the epidemics at macroscale.

4.7.6 Inferring Synthetic Networks

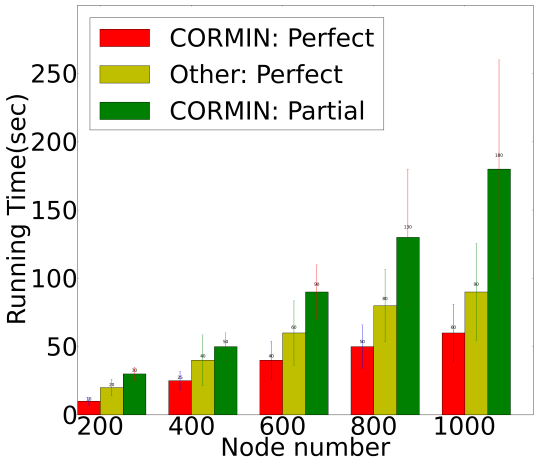
CORMIN performs consistently better than the existing methods on inferring synthetic networks grown via different growth models from different diffusion models as seen in Table 4.5. Scores in bold represent the cases where *CORMIN* performs reasonably better than the existing methods. *CORMIN* performs significantly better than the existing methods on inferring graphs grown via FF and LPA. All methods perform similar on inferring RDS networks, and they perform the worst on inferring DMC networks. This lower performance can be explained by the loopy structure of DMC networks. In general, *CORMIN* can easily achieve $F1$ score greater than 0.5 in all models except DMC. Table 4.5 shows the performance only from 250 traces but *CORMIN*'s performance is consistent across different number of traces and conditions.

4.7.7 Scalability and Performance under Other Challenging Cases

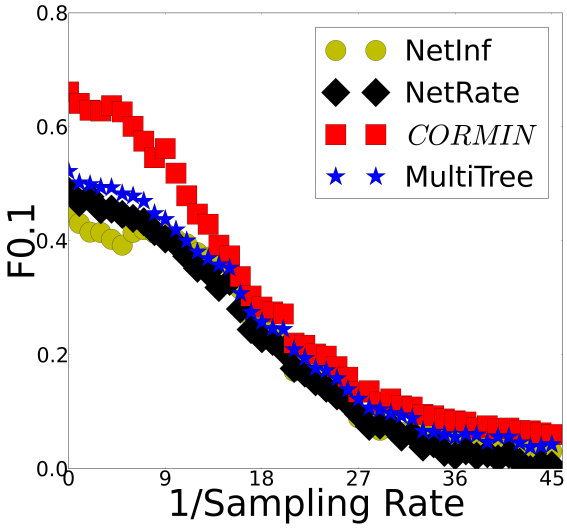
CORMIN infers graphs faster than the existing methods when the data is perfect as in Figure 4.6a which shows the mean running time as well as the standard deviation from 20 runs over graphs of different sizes on a single CPU computer. It runs slower when the diffusion data is partial, but this running time is still reasonable considering it is capable of modeling the probabilistic data, and it can infer networks better than the existing methods. *CORMIN* infers *Contact-static* in less than a minute from 500 traces on a personal laptop.

CORMIN is also scalable to very large graphs since convex risk minimization can be done independently for each node. For instance, *CORMIN* can optimally infer graphs having hundred thousands of nodes in less than 3 minutes by using 100 processors since it can be parallelized without losing the optimality of the relaxation.

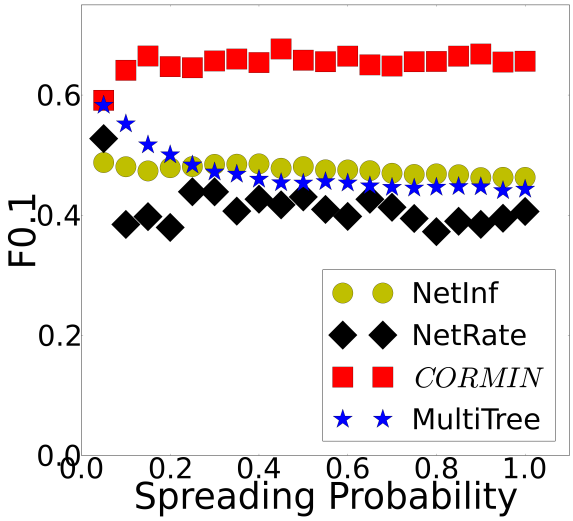
CORMIN infers *Contact-static* better than the existing methods when the data is undersampled as in Figure 4.6b. In this plot, x axis shows the inverse of the sampling rate; 0 corresponds to the perfectly known case, and y means we only observe 1 time point in each y-length interval.



(a) Running Time



(b) Affect of sampling rate



(c) Affect of spreading probability s_{uv}

Figure 4.6: a) Comparison of running time of *CORMIN* and the existing methods, b) $F0.1$ vs. $\frac{1}{\text{Sampling Rate}}$ from 250 traces over *Contact-static*, c) Affect of spreading probability s_{uv} on inferring *Contact-static* from 250 traces

Diffusion data sampled at a lower rate provides less information, but *CORMIN* can tolerate such missing information up to a certain sampling rate as it is still more accurate than the existing methods. However, at sampling rates lower than $\frac{1}{16}$, all methods start to perform similarly and worse since almost all the diffusion information is lost. *CORMIN* is more robust to noise as shown previously in Figure 4.3c, and it consistently performs the best under different probability of diffusion (s_{uv}) parameters as in Figure 4.6c.

4.8. Conclusion

In this research, we present a convex risk minimization based approach to infer unknown graphs under SEIR models from probabilistic, partially observable diffusion data. We show improved graph recoverability under both uncertain and perfect node states at multiple scales; our method is capable of recovering the influenza transmission network at microscale and transmission rates at macroscale. The performance advantage of our method can be explained by its better modeling of both edge existence and nonexistence from diffusion data, better handling uncertain data, bounding the number of edges by using covering constraints for perfectly known diffusion data, and its ability to formulate the inference problem at multiple scales. We believe that our model-based inference method can also be extended to the other similar biological network inference problems.

Chapter 5

Diffusion Archaeology: Reconstructing The Diffusion History From Present-Day Data

A preliminary version of this chapter appeared in IEEE 14th International Conference on Data Mining ICDM 2014 with the title *Diffusion Archaeology for Diffusion Progression History Reconstruction* [127].

5.1. Introduction

Dynamic processes over networks are used to model and analyze properties of various social and biological systems. Diffusion is special case of those processes in which a spread (e.g., an infection) starts from some part of the graph and spreads to other portions over time via edges of the graph. Some examples are virus propagation in computer networks [130], and idea and gossip spreading in social networks [157]. A diffusion model, such as the commonly studied SIRS and SEIRS models, defines the set of possible states that the nodes of the graph can be in and rules for probabilistically switching between states. Recently, [112] introduced the *VPM* (Virus Propagation Model) that generalizes all those Markovian diffusion models and defines the hierarchical relationships between them.

It is not always easy to know the whole diffusion progression, initial diffusion conditions, or the time it started due to several limitations. For example, existence of a computer virus diffusion over the computer network may only be noticed after a significant number of computers stop operating. A similar problem exists in detecting influenza diffusion [121]. We may also not track the diffusion of a virus in email networks and a contaminant in a water distribution network due to privacy and physical limitations, respectively. In all these cases, it is essential to learn more about the past to take precautions to prevent future epidemics, to learn more about the true diffusion mechanics, to provide safer water, to break privacy and so on. However, given present-day diffusion data, it is not trivial to search for the most likely diffusion progression in the past since there will be many valid histories leading to the observed data.

In this research, we tackle this problem of inferring a complete diffusion history from one

or more diffusion snapshots for discrete-time SEIRS-type diffusion models that include SI, SIS, SIR, SIRS, SEIR, SEIS, SEIRS. Those models with their abstract states and independent cascade (IC) assumption [73] have been used to model many forms of diffusion in many different domains [91, 121, 157].

Complete diffusion history reconstruction has not been previously studied but similar problems exist in the literature. The most relevant such problem is *Initial Spreader Identification* where we want to identify the most probable initial infected nodes that started a diffusion. Among approaches for this problem, *Keffectors* [85] identifies the k best-possible initial spreaders. However, it requires an estimate of the number of initial spreaders to be given as input. *Rumor* [132] finds the most probable spreader by estimating the rumor centrality of each node but it assumes a single initial spreader, and it is only defined for the SI model. Lastly, *NetSleuth* [113] lets multiple nodes be initial spreaders without requiring this number as an input parameter. However, it works only for the restricted cases of SI model, and it is based on a MDL heuristic without any provable performance guarantee. None of these methods infer the whole diffusion progression, as our approaches do. Another related problem is *Graph Inference* where we want to reconstruct the unknown graph from observed multiple diffusion traces over it. This problem is fundamentally different than the history reconstruction problem as graph inference methods [57, 58] search in the graph space assuming full observability of multiple traces whereas our methods search in temporal diffusion progression space as they try to complete the missing history of a single trace.

We formulate the diffusion history reconstruction problem as that of determining the maximum likelihood (ML) history given diffusion snapshots that may come from multiple time points. We designed an algorithm called *DHR-sub* (submodular history reconstruction on discrete dynamics) that reconstructs the history before the earliest measured time point by greedily maximizing the non-monotone submodular log-likelihood at each previous time step. It further reconstructs the history between the consecutive diffusion data time points by solving the problem as non-monotone submodular maximization under matroid base constraints.

Though accurate and practical for smaller graphs, *DHR-sub* can take some time to solve. To reconstruct diffusion history faster, we designed *DHR-pcdsvc* that solves the first-order Taylor approximation relaxation of the log-likelihood. We define this new problem as *Prize-Collecting Dominating-Set Vertex Cover*, and show that it can be approximated within a factor of $O(\log(|V|))$. This problem can be further relaxed by removing the covering constraints; it becomes *Prize-Collecting Vertex Cover*, and we design *DHR-pcvc* that approximates it by a factor of 2 for non-bipartite models and solves this newer relaxation optimally by transforming it to s-t mincut for bipartite models. We also design ensemble approaches for all of our methods that estimate the robust set of initial spreaders from multiple runs of the algorithm.

In summary, our main contributions are:

- Our methods reconstruct the whole diffusion history nonparametrically for all SEIRS type models whereas the existing methods only identify the initial spreaders for certain models;
- Our methods formulate the problem in terms of diffusion likelihood, and we give some performance guarantees on the quality of the obtained solutions;
- Our relaxation methods *DHR-pcdsvc* and *DHR-pcvc* scale well to history reconstruction over tens of thousands of nodes with provable performance guarantees;
- Our methods reconstruct the history better by using the diffusion information from all the

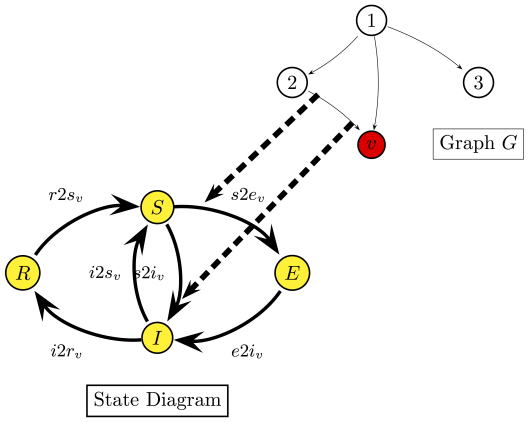


Figure 5.1: SEIRS state transition diagram

nodes (not just from the infected nodes), and from multiple time points if available;

- We use reconstructed histories to predict several diffusion features such as speed and acceleration that are not apparent in the observed portion of the diffusion.

Our methods more accurately identify initial spreader sites on a water distribution network and on simulated networks. In terms of history reconstruction, we compared our methods with a baseline heuristic since there is no previous method. All our methods can accurately reconstruct several meme diffusion histories on blog networks. They also perform better on synthetic networks under different models. In general, all our methods reconstruct the diffusion history reasonably fast and accurately compared to the hardness of the problem (see Section 5.7.6). In many cases, relaxations of the original problem can reconstruct the diffusion history almost as good as the original formulations in a far shorter amount of time. Lastly, we also estimate the speed and acceleration dynamics of several memes over blog network from their reconstructed histories. In this case, estimated dynamics from quite a few diffusion snapshots match the true dynamics almost perfectly producing decent whole history reconstruction performance. Overall, our results for different types of diffusion show that many characteristics of complete diffusion history can be inferred with proper modeling and methods.

5.2. SEIRS Diffusion Dynamics

The SEIRS diffusion dynamics over directed graph $G = (V, E)$ with possible state transitions is shown in Figure 5.1. The SEIRS states are Susceptible (*S*), Exposed but not contagious (*E*), Infected and contagious (*I*), and previously infected but now Recovered (or immune to the infection) (*R*). Those states are general enough abstractions to model various forms of diffusion in different contexts [91, 121]. For instance, the infected state models people having influenza symptoms in influenza diffusion over humans, and it represents creation of a blog entry about a topic in idea diffusion. Similarly, the recovered state could represent recovery of a person from influenza or the decontamination of a water tower from chemical contaminants depending on the context.

Symbol	Definition and Description
$G = (V, E)$	directed graph G
$P(v)$, $S(v)$	set of predecessors, successsors of node v
$\mathcal{S}$	SEIRS states (*S*,*E*,*I*,*R*)
$\mathcal{M}$	diffusion model of SEIRS type models
p_{uv}	probability of diffusion from infected node u to susceptible node v
$e2i_v$, $i2r_v$, $r2s_v$, $s2e_v$, $s2i_v$, $i2s_v$	probability of ($E \rightarrow I$, $I \rightarrow R$, $R \rightarrow S$, $S \rightarrow E$ $S \rightarrow I$, $I \rightarrow S$) transition for v
t_v^s, t_v^e, t_v^i, t_v^r	time v transitions into (*S*, *E*, *I*, *R*)
S_t, E_t, I_t, R_t	set of nodes that are in (*S*, *E*, *I*, *R*) at time t
D_t	diffusion snapshot at time t
D	given diffusion data
l_D	length of diffusion D
T_D	ordered set of time points of D
t_{min}, t_{max}	$\min(T_D)$, $\max(T_D)$
f_{min}, f_{max}	$\frac{\min(T_D)}{l_D}$, $\frac{\max(T_D)}{l_D}$

Table 5.1: Table of Symbols

In SEIRS model, diffusion starts at time $t = 0$ from set of initially infected nodes and progresses over G in discrete time steps. Let S_t, E_t, I_t, R_t be the set of *S*, *E*, *I*, *R* nodes at time t respectively. At each time step, infected nodes spread the infection to the susceptible nodes with certain probability. This $S \rightarrow E$ transition is exogenous; it is affected by G and probability of exogenous transition for susceptible node v at time t is $1 - \prod_{u \in P(v) \cap I_t}(1 - p_{uv})$, where $P(v)$ is the set of nodes with edges into v and p_{uv} is the probability of transmission of the agent over edge (u, v). The remaining $E \rightarrow I$, $I \rightarrow R$, $R \rightarrow S$ transitions are endogenous; their transition probabilities are $e2i_v$, $i2r_v$, $r2s_v$ respectively, and they are not affected by G. For every node at each time step, if a transition succeeds, the node transitions to a new state. Otherwise, it follows similar procedure at next time step, independent of the previous trials. SEIRS type models are Markovian since state of a node at time t depends on its state and its neighbors' states at previous time steps, and it obeys independent cascade (IC) [73] assumption which states that a diffusion from one of nodes predecessor is enough for node to become exposed/infected. The symbols used in this text are given in Table 5.1 for reference.

SEIRS-type models include the well-known SI, SIR, SIS, SIRS, SEIR, SEIRS models [61]. SEIRS is the most general model among these models, and some of its transitions disappear or change slightly in other models. For instance, in SIR, there is no exposed state; the exogenous transition is $S \rightarrow I$ since nodes proceed directly to the infected state, and there is no $R \rightarrow S$ transition. We can classify SEIRS type models in various ways. SIRS, SEIRS are *loopy* models where $R \rightarrow S$ transition is available whereas SI, SIR, SEIR are *non-loopy* models. We can also split SEIRS type models into *bipartite* and *non-bipartite* models: a node that gets the infection directly transitions into infected state in non-bipartite models such as SI, SIR, SIRS, SIS whereas it goes through the exposed state for bipartite models such as SEIR, SEIRS. The model may also

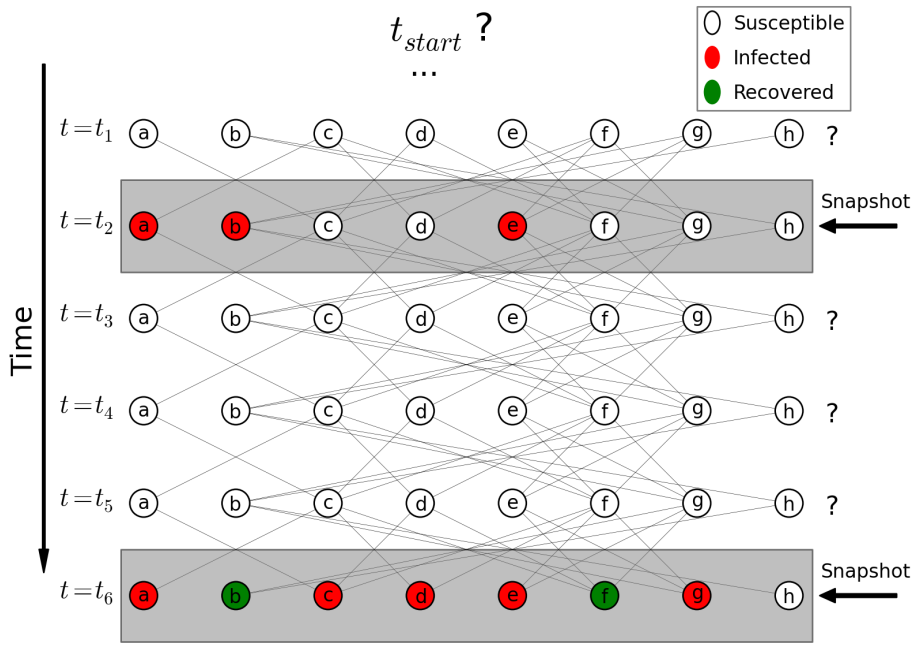


Figure 5.2: Example Problem: SIR Diffusion over 8 node graph where we can only observe t_2 and t_6 without knowing the initial diffusion time t_{start}. We want to reconstruct the missing diffusion snapshots from t_{start} onwards

be either *uniform* in which case all of the transition probabilities are the same for each edge and node, or *non-uniform* in which case the probabilities may vary over the edges and nodes. We discuss the general non-uniform case here; the uniform case is a simple specialization.

5.3. Diffusion History Reconstruction Problem

For diffusion D, let $D_t = (S_t, E_t, I_t, R_t)$ be the state of the nodes at the time t, where S_t is the set of susceptible nodes, etc. D_t is a *diffusion snapshot*. We define Problem 2 to reconstruct the diffusion history when the diffusion length is unknown:

Problem 2. *We are given: a graph $G = (V, E)$, state transition probabilities (p_{uv}, $e2i_v$, $i2s_v$, $i2r_v$, $r2s_v$) that define an SEIRS-type model, a collection of time points T_D at which snapshots were taken, and a collection of diffusion snapshots $D = \{D_t\}$ for $t \in T_D$. Each snapshot records the state of every node at a single time point, partitioning them into $V = S_t \cup E_t \cup I_t \cup R_t$.*

Our goal is to infer the past states (susceptible, exposed, infected and recovered) of every node at every time $t \notin T_D$.

Figure 5.2 illustrates the history reconstruction problem for the SIR model. Each subsequent layer shows the progression of time, and we want to reconstruct the diffusion progression from unknown initial time t_{start} onwards given full state knowledge at subset of time points. The *initial spreader identification problem* is special case of Problem 2 where we want to identify only the initial *infected* nodes.

5.4. Non-monotone Submodular History Reconstruction (*DHR-sub*)

Let $t_{max} = \max(T_D)$, $t_{min} = \min(T_D)$, t_{start} be the unknown initial diffusion time, and $\mathcal{M}$ be the specific SEIRS type model that diffusion snapshots are collected over. SEIRS type models are Markovian so the probability of diffusion $D = \{D_{t_{start}}, \ldots, D_{t_{max}}\}$ that starts at t_{start} and progresses until t_{max} can be written as the multiplication of the probability of each time step in terms of previous time steps as in (5.1):

$$P(D) = \prod_{j=t_{start}+1}^{t_{max}} P(D_j | D_{j-1}, \ldots, D_{t_{start}})\ P(D_{t_{start}}) \tag{5.1}$$

We assume the state transition probabilities (p_{uv}, $e2i_v$, $i2r_v$) to be same at each time step, so the overall diffusion probability in (5.1) simplifies to Equation (5.2) under this memoryless property:

$$P(D) = \prod_{j=t_{start}+1}^{t_{max}} P(D_j | D_{j-1})\ P(D_{t_{start}}) \tag{5.2}$$

Let T_D be the ordered set of observed time points, $X_j = (S_j, E_j, I_j, R_j)$ be the unknown state knowledge at time $\forall j \notin T_D$, and $X = \{X_j : j \in \{t_{start}, \ldots, t_{max}\} \setminus T_D\}$. Given a collection of diffusion snapshots D for T_D, our goal is to reconstruct the most probable diffusion progression (X) by maximizing the log-likelihood as in Equation (5.3)–(5.5):

$$\underset{X}{\text{argmax}}\ \log(\mathcal{L}(X|D)) = \underbrace{\log(\mathcal{L}^{pre})}_{\textbf{\textit{DHR-sub-early}}} + \sum_{(j,k)\in\mathcal{P}(T_D)} \underbrace{\log(\mathcal{L}^{in}_{j,k})}_{\textbf{\textit{DHR-sub-between}}} \tag{5.3}$$

$$\text{s.t. } \textbf{IntraConsistent}(X_j, \mathcal{M}),\ j \in t_{start}, \ldots, t_{max} - 1 \tag{5.4}$$

$$\textbf{InterConsistent}(X_j, X_{j+1}, \mathcal{M}),\ j \in t_{start}, \ldots, t_{max} - 1 \tag{5.5}$$

where $\mathcal{L}^{pre}$ and $\mathcal{L}^{in}_{j,k}$ are defined in (5.6)–(5.7), $\mathcal{P}(T_D) = \{(t_j, t_{j+1}),\ j \in 1, \ldots, |T_D| - 1\}$, and maximum log-likelihood estimate is the same as the maximum likelihood estimate since the logarithm is a monotonically increasing function:

$$\mathcal{L}^{in}_{j,k} = P(X_{j+1}|D_j)\ P(D_k|X_{k-1}) \prod_{t\in j+1,..,k-2} P(X_{t+1}|X_t) \tag{5.6}$$

$$\mathcal{L}^{pre} = \prod_{j=1}^{t_{min}-t_{start}} P(X_{t_{min}-j+1}|X_{t_{min}-j})\ P(X_{t_{start}}) \tag{5.7}$$

There are two types of constraints: **IntraConsistency** constraints (5.4) make sure that the variable assignments at each time step are valid under $\mathcal{M}$: every node belongs to a single state at each j, and **InterConsistency** constraints (5.5) make sure that the diffusion between each pair of consecutive time steps is valid according to rules of $\mathcal{M}$: every node that got the infection at j has at least one infected predecessors at $j-1$, and node transitions are valid according to $\mathcal{M}$.

For instance, recovered nodes cannot become susceptible if $\mathcal{M}$ is not loopy. These constraints are described in more detail below.

The diffusion history between consecutive observed T_D pairs is independent of each other since Equation 5.2 is memoryless, and each D_j completely describes states of all nodes at time j. Thus, maximizing (5.3)–(5.5) can be partitioned into multiple independent subproblems of two types that can be optimized independently. The first type maximizes $\log(\mathcal{L}^{pre})$ under the consistency constraints to reconstruct the history before t_{min} (***DHR-sub-early***). The second type maximizes $\log(\mathcal{L}^{in}_{j,k})$ to reconstruct the history between the snapshots from time j and time k under the consistency constraints (***DHR-sub-between***). We define algorithms for both types of subproblems below. In the text, we use D_j and X_j interchangeably for $\forall j \in T_D$.

5.4.1 History reconstruction before the earliest observed snapshot (*DHR-sub-early*)

To find the most likely diffusion history before t_{min}, we solve the problem:

$$\underset{X}{\text{argmax}}\ \log(\mathcal{L}^{pre}) = \sum_{j=t_{start}}^{t_{min}-1} \log\Big(P(X_{j+1}|X_j)\Big) + \log\Big(P(X_{t_{start}})\Big) \tag{5.8}$$

$$\text{s.t. } \textbf{IntraConsistent}(X_j, \mathcal{M}),\ j \in t_{start}, \ldots, t_{min}-1 \tag{5.9}$$

$$\textbf{InterConsistent}(X_j, X_{j+1}, \mathcal{M}),\ j \in t_{start}, \ldots, t_{min}-1 \tag{5.10}$$

We assume a uniform prior $P(X_{t_{start}})$ over set of initially infected nodes since we do not have any extra information about them. We now discuss how to formulate the objective function and constraints above in terms of binary variables representing each node's state.

Expressing the objective function (5.8)

Given X_j, the probability of observing the diffusion snapshot X_{j-1} at time $j-1$ can be expressed as:

$$P(X_j|X_{j-1}) = \mathcal{L}(X_{j-1}|X_j) = \mathcal{L}^j_s\ \mathcal{L}^j_e\ \mathcal{L}^j_i\ \mathcal{L}^j_r \tag{5.11}$$

where $\mathcal{L}^j_s$, $\mathcal{L}^j_e$, $\mathcal{L}^j_i$, $\mathcal{L}^j_r$ are the likelihoods of the nodes in S_j, E_j, I_j, R_j states respectively in terms of X_{j-1}.

To define $\mathcal{L}^j_s$, $\mathcal{L}^j_e$, $\mathcal{L}^j_i$, $\mathcal{L}^j_r$, we introduce a single binary variable for each node to define its state at time $j-1$ given its state at time j. A binary variable is sufficient because there are only two possibilities for a node at time $j-1$ given its state at time j: either the node is in the same state as time j, or the node has made a state transition at time j, and when computing $\mathcal{L}(X_{j-1}|X_j)$, the state at time j is known.

We define a variable $s_{v,j-1}$ for every node $v \in E_j$, and $r_{v,j-1}$ for every node $v \in R_j$. For every node $v \in I_j$, we define variable for the incoming state of *I* ($e_{v,j-1}$ for bipartite $\mathcal{M}$ and $s_{v,j-1}$ for non-bipartite $\mathcal{M}$). Similarly, for every node $v \in S_j$, we define a variable for the incoming state of *S* if $\mathcal{M}$ is loopy, otherwise we do not need to define the variable, since we know that if a node is in S_j it must be in S_{j-1}.

With these variables, the likelihoods in (5.11) are explicitly defined as in (5.12)–(5.15) for SEIRS model as:

$$\mathcal{L}_e^j = \prod_{v \in E_j} \left(\mathcal{L}_{e2e}^{v,j-1} \left(1 - \mathcal{L}_{s2e}^{v,j,I} \, \mathcal{L}_{s2e}^{v,j,R}\right)^{s_{v,j-1}} \right) \tag{5.12}$$

$$\mathcal{L}_s^j = \prod_{v \in S_j} \left(\left(\mathcal{L}_{s2e}^{v,j,I} \, \mathcal{L}_{s2e}^{v,j,R} \right)^{s_{v,j-1}} (r2s_v)^{1-s_{v,j-1}} \right) \tag{5.13}$$

$$\mathcal{L}_r^j = \prod_{v \in R_j} \left((i2r_v)^{1-r_{v,j-1}} (1 - r2s_v)^{r_{v,j-1}} \right) \tag{5.14}$$

$$\mathcal{L}_i^j = \prod_{v \in I_j} \left(\left((e2i_v)^{e_{v,j-1}} (1 - i2r_v)^{1-e_{v,j-1}} \right) \right) \tag{5.15}$$

where the sub-terms are defined as:

$$\mathcal{L}_{s2e}^{v,j,I} = \prod_{u \in P(v) \cap I_j} (1 - p_{uv})^{1-e_{u,j-1}}$$

$$\mathcal{L}_{s2e}^{v,j,R} = \prod_{u \in P(v) \cap R_j} (1 - p_{uv})^{1-r_{u,j-1}}$$

$$\mathcal{L}_{e2e}^{v,j} = \prod_{v \in E_j} (1 - e2i_v)^{1-s_{v,j-1}}$$

Each likelihood above for a given state ($\mathcal{L}_s^j$, $\mathcal{L}_e^j$, $\mathcal{L}_i^j$, $\mathcal{L}_r^j$) has two parts: the likelihood of the nodes staying at the given state, and the likelihood of the nodes transitioning towards the given state. For example, $\mathcal{L}_e^j$ is the likelihood for nodes $v \in E_{j-1}$ not to transition to infected state at time j, and nodes $v \in S_{j-1}$ to become exposed at time j. This gives us an explicit definition of objective function (5.8) in terms of a collection of binary variables.

Likelihoods (5.12)–(5.15) are defined for the most general model SEIRS, some of the likelihood terms disappear, or change slightly for models that are missing some of the states. For instance, parts including $r2s_v$ in $\mathcal{L}_s^j$ and $\mathcal{L}_r^j$ disappear for non-loopy $\mathcal{M}$, the likelihood representing the exogenous transition $\mathcal{L}_{s2e}$ is replaced by the similarly defined $\mathcal{L}_{s2i}$ for non-bipartite models, etc.

Expressing the constraints in equations (5.9) **and** (5.10)

The intra-consistency constraints (5.9) that require every node have a single state at each time step are already implied by the objective function since there is only single variable for every node modeling the two possibilities. The inter-consistency constraints (5.10) can be modeled as packing constraints:

$$\sum_{u \in P(v) \cap I_j} e_{u,j-1} + \sum_{u \in P(v) \cap R_j} r_{u,j-1} + s_{v,j-1} \leq d_v, \forall v \in E_j \tag{5.16}$$

These constraints make sure every node that became exposed at time t ($v \in E_j, S_{j-1}$) has at least one incoming edge from node $u \in I_{j-1}$ ($u \in I_j \cup R_j$). However, these constraints (5.16) are already represented in the objective function (5.11) since the higher order term $\log(\mathcal{L}_e^j)$ takes the lowest possible value $\log(0) = -\infty$ when any of them are not satisfied.

Optimizing the likelihood under these constraints

Since t_{start} is unknown, we reconstruct the history using the above likelihoods and constraints by iteratively maximizing the likelihood at each time step $t_{start} \leq j < t_{min}$ backwards starting from $t_{min} - 1$, where the state is known. In each iteration, given X_j, we reconstructs the states at the previous time step $j-1$ (X_{j-1}) by maximizing:

$$\text{max F} = \log(\mathcal{L}_s^j) + \log(\mathcal{L}_e^j) + \log(\mathcal{L}_i^j) + \log(\mathcal{L}_r^j) \tag{5.17}$$

Objective F for single step reconstruction is submodular as proven in Theorem 5.4.1 for all SEIRS type models except SIS. For SIS model, history reconstruction can still be expressed as submodular maximization under packing and partition matroid constraints by modifying F as in Theorem 5.4.2.

Theorem 5.4.1. *F in Equation* (5.17) *is non-monotone submodular for all SEIRS type models except SIS.*

Proof. F has three types of terms; higher order terms from $\log(\mathcal{L}_e^j)$, quadratic or linear terms from $\log(\mathcal{L}_s^j)$ depending on $\mathcal{M}$ and linear terms from $\log(\mathcal{L}_i^j)$ and $\log(\mathcal{L}_r^j)$. F is non-monotone since linear and quadratic terms are either positive or negative depending on $\mathcal{M}$, transition distribution parameters and the terms from $\log(\mathcal{L}_s^j)$ that model the probability of susceptible nodes not being infected/exposed.

F is submodular when $F(A+x) - F(A) \geq F(B+x) - F(B)$ for every $A \subset B$ and for every $x \in U \setminus (A \cup B)$. To prove submodularity of F, we prove the submodularity of each term in F since summation of submodular functions is also submodular. Linear terms of F are unimodular, so they are submodular. Quadratic terms show up in $\log(\mathcal{L}_s^j)$ when $\mathcal{M}$ is loopy and when the model is not SIS, each quadratic term is one of the following: $Q(r_{v,j-1}, r_{u,j-1}) = \log(1-p_{uv})(1-r_{v,j-1})(1-r_{u,j-1})$, $Q(r_{v,j-1}, e_{u,j-1}) = \log(1-p_{uv})(1-r_{v,j-1})(1-e_{u,j-1})$ or $Q(r_{v,j-1}, i_{u,j-1}) = \log(1-p_{uv})(1-r_{v,j-1})(1-i_{u,j-1})$. All those terms are submodular since they satisfy the inequality $Q(0,0) + Q(1,1) \leq Q(0,1) + Q(1,0)$.

Then, we need to prove the submodularity of the higher-order terms that depend on G to prove submodularity of F. Higher-order terms appear in either $\log(\mathcal{L}_e^j)$ for bipartite models or $\log(\mathcal{L}_i^j)$ for non-bipartite diffusion models. Depending on $\mathcal{M}$, we need to prove either $T = s_{v,j-1} \log\left(1.0 - \mathcal{L}_{s2e}^{v,j,I}\ \mathcal{L}_{s2e}^{v,j,R}\right)$ or $T = s_{v,j-1} \log\left(1.0 - \mathcal{L}_{s2i}^{v,j,I}\ \mathcal{L}_{s2i}^{v,j,R}\right)$ is submodular. Each variable might appear at two positions of T; either inside or outside the logarithm. When $\mathcal{M}$ is bipartite, each variable can only appear in one of those positions whereas it can appear in both positions for non-bipartite $\mathcal{M}$. Let $V_e = \bigcup_{u \in P(v) \cap I_j} e_{u,j-1}$, $V_r = \bigcup_{u \in P(v) \cap E_j} r_{u,j-1}$, x be the variable to be added, X be the current set of added variables, $K = \Pi_{V_e \cup V_r}(1-p_{uv})$ and $P_t = (1-p_{tv})^t$ for every $t \in V_e \cup V_r$, T is submodular as proven below.

- If x is outside the logarithm, let $A = \{a, b\}$ and $B = \{a, b, c\}$. Then, $T(A+x) = \log\left(1 - \frac{K}{P_a P_b P_x}\right)$, $T(B+x) = \log\left(1 - \frac{K}{P_a P_b P_c P_x}\right)$ and $T(A+x) - T(A) \geq T(B+x) - T(B)$ will be satisfied since $T(A+x) \geq T(B+x)$ and $T(A) = T(B) = 0$.
- If x is inside the logarithm, when $s_{v,j-1} \notin X$, submodularity is trivially satisfied since $T(A) = T(A+x) = T(B) = T(B+x) = 0$. When $s_{v,j-1} \in X$, let $A = \{a\}$ and $B = \{a, c\}$ ($A \subset B$), submodularity is satisfied as shown in Equation (5.18)–(5.20).

$$T(A+x)-T(A) \geq T(B+x)-T(B) \tag{5.18}$$

$$\log\left(\frac{1-\frac{K}{P_aP_x}}{1-\frac{K}{P_a}}\right) \geq \log\left(\frac{1-\frac{K}{P_aP_bP_x}}{1-\frac{K}{P_aP_c}}\right) \tag{5.19}$$

$$KP_aP_b(1-P_b)(1-P_a) \geq 0 \tag{5.20}$$

Then, F is submodular since each summation term including the higher-order ones is submodular.

□

Theorem 5.4.2. *Program* (5.17)–(5.16) *for SIS can be expressed as submodular maximization under both packing and partition matroid constraints.*

Proof. Quadratic terms $Q(s_{v,j-1}, s_{u,j-1})$ from $\mathcal{L}_s^j$ are supermodular for SIS but they can be turned into submodular ones as follows: We define new varible $i_{v,j-1}$ for every node $v \in \{S_j \cup I_j\}$ to represent whether v is infected at time $j-1$. Then, we obtain the new objective function F^* by replacing each supermodular $Q(s_{v,j-1}, s_{u,j-1}) = \log(1-p_{uv})s_{v,j-1}(1-s_{u,j-1})$ with $Q^*(s_{v,j-1}, s_{u,j-1}) = \log(1-p_{uv})s_{v,j-1}i_{v,j-1}$. We also add assignment constraints of $s_{v,j-1} + i_{v,j-1} = 1$ for every node $v \in \{S_j \cup I_j\}$ to make sure node v is either infected or susceptible at $j-1$. Each $Q^*(s_{v,j-1}, s_{u,j-1})$ in F^* is submodular since it satisfies the inequality $Q^*(0,0) + Q^*(1,1) \leq Q^*(0,1) + Q^*(1,0)$. Then, F^* is submodular since the rest of the higher-order terms are submodular as proven in Theorem 5.4.1. Assignment constraints define partition matroid and the problem of reconstructing history at time $j-1$ becomes submodular maximization under both partition matroid and existing packing constraints for SIS model.

□

Therefore, optimizing (5.17) is a non-monotone submodular maximization problem. Non-monotone submodular maximization is NP-hard since its special cases such as *MAX DICUT* is NP-hard [46]. To solve this problem, we apply the deterministic non-monotone submodular maximization method by [47] repeatedly between adjacent time steps and iterate until the estimates between the consecutive time steps are same, indicating that we have reached the initial t_{start} state. At each step, [47] maximizes a normalized F_n at every step that is obtained by adding $-F(\emptyset)$ to every $S \subset 2^N$ so $F_n(\emptyset) = 0$, where $F(\emptyset)$ is the value of the objective if no nodes change states between adjacent time points. As applied here, this is done by starting with an initial solution $X_j' = \emptyset$ that represents the same state assignments between the consecutive time steps j and $j-1$. For each time step, we add the node with the most increase in F_n to the set of nodes that have changed state. Algorithm 1 gives a schematic outline of the procedure.

The method by [47] has a $\frac{1}{3}$ approximation ratio for normalized submodular functions, and we found it to perform in practice better than the randomized algorithm [19] with approximation ratio 0.5 due to the structure of our problem. The $\frac{1}{3}$ ratio for normalized F_n implies a data-dependent bound for F as proven in Theorem 5.4.3 where $F(\emptyset) = -S_0$, X_{opt} is the set of elements maximizing F and $F(X_{opt}) = -O$.

Theorem 5.4.3. *Algorithm 1 has approximation guarantee of* $k + \frac{S_0}{O}(1-k)$ *for* $k = \frac{1}{3}$ *in terms of minimization of supermodular* $-F$ *for each of its iteration.*

Algorithm 1 *DHR-sub-early*

$j \leftarrow t_{min} - 1$
repeat
  $X'_j \leftarrow \emptyset$ ▷ X'_j is the set of nodes that changed state at time j
  repeat
    Add nodes to X'_j according to the rule for non-monotone submodular maximization approximation [47]
  until no node can be added that improves the score.
  $j \leftarrow j - 1$
until $X'_j = X'_{j+1}$

Proof. Let X be the set of elements returned by the non-monotone submodular maximization algorithm and $F(X) = -M$. We are interested in upper-bounding the supermodular minimization ratio ($\frac{M}{O}$) for $-F$. Since F_n is obtained by adding S_0 to each set in F, $\frac{F_n(X)}{F_n(X_{opt})} = \frac{S_0 - M}{S_0 - O} \geq k$ and we obtain $\frac{M}{O} <= k + \frac{S_0}{O}(1-k)$. Here, $\frac{S_0}{O}(1-k)$ makes the approximation ratio data-dependent and this ratio is the best we can achieve when k is tight for non-monotone submodular maximization. This data-dependent bound is also the best we can achieve in terms of supermodular minimization perspective since non-negative supermodular minimization problem cannot be approximated in constant factor unless $P = NP$ [155].

□

5.4.2 History Reconstruction Between Consecutive Snapshots (*DHR-sub-between*)

History reconstruction for every interval between by consecutive, observed T_D pairs is independent of other intervals. Therefore, we can solve each independently by solving the following problem:

$$\underset{X}{\text{argmax}}\ T = \log(P(X_{j+1}|D_j)) + \log(P(D_k|X_{k-1})) + \sum_{t \in j+1,..,k-2} \log(P(X_{t+1}|X_t)) \tag{5.21}$$

$$\text{s.t } \textbf{IntraConsistent}(X_t, \mathcal{M}),\ t \in j+1, \ldots, k-1 \tag{5.22}$$

$$\textbf{InterConsistent}(X_t, X_{t+1}, \mathcal{M}),\ t \in j, \ldots, k-1 \tag{5.23}$$

where we only consider X_t that lay within the $[j, k]$ interval bracketed by diffusion snapshot observations D_j and D_k

Expressing objective (5.21)

Objective (5.21) has three parts: $\log(P(D_k|X_{k-1}))$ is same as the single step backwards reconstruction of *DHR-sub-early*, and $\log(P(X_{j+1}|D_j))$ is a trivial forward diffusion expression with unknown X_{j+1} and known D_j. On the other hand, both X_{t+1} and X_t are unknown in

$\log(P(X_{t+1}|X_t))$. This expression can be written explicitly as:

$$\log(P(X_{t+1}|X_t)) = \log(\mathcal{L}_s^{t+1}) + \log(\mathcal{L}_e^{t+1}) + \log(\mathcal{L}_i^{t+1}) + \log(\mathcal{L}_r^{t+1}) \tag{5.24}$$

where the likelihoods are defined as:

$$\mathcal{L}_e^{t+1} = \prod_{v \in V} ((1 - e2i_v)^{e_{v,t}\, e_{v,t+1}} \ (1.0 - \mathcal{L}_{exo})^{e_{v,t+1}}) \tag{5.25}$$

$$\mathcal{L}_s^{t+1} = \prod_{v \in V} (\mathcal{L}_{exo}^{s_{v,t+1}} (r2s_v)^{r_{v,t}\, s_{v,t+1}}) \tag{5.26}$$

$$\mathcal{L}_r^{t+1} = \prod_{v \in V} \left((i2r_v)^{i_{v,t}\, r_{v,t+1}} (1.0 - r2s_v)^{r_{v,t}\, r_{v,t+1}}\right) \tag{5.27}$$

$$\mathcal{L}_i^{t+1} = \prod_{v \in V} \left((e2i_v)^{e_{v,t}\, i_{v,t+1}} (1.0 - i2r_v)^{i_{v,t}\, i_{v,t+1}}\right) \tag{5.28}$$

and the term in (5.25) is:

$$\mathcal{L}_{exo} = \prod_{u \in P(v)} (1 - p_{uv})^{i_{u,t}\, s_{v,t}}$$

Objective (5.21) is non-monotone submodular as proven in Theorem 5.4.4.

Theorem 5.4.4. *T in (5.21) is non-monotone submodular for all SEIRS type models.*

Proof. We prove the submodularity of $\log(\mathcal{L}_{j,k}^{in})$ by proving the submodularity of each of its summation terms. $\log(P(X_{j+1}|D_j))$ estimates the most probable diffusion snapshot at $j+1$ given D_j. It is a forward estimate and if we use the same variable naming as in Section 5.4.1, it becomes a linear function of X_{j+1} and thus submodular.

$\log(P(D_k|X_{k-1}))$ is same as F (5.17) in Section 5.4.1 and it is submodular as proven in Theorem 5.4.1.

Every $\log(P(X_{t+1}|X_t))$ involves the variables from both time steps t and $t+1$. Here, we do not know the exact node states at both time steps so we define all possible state variables for every node for both time steps ($s_{v,t}$, $e_{v,t}$, $i_{v,t}$, $r_{v,t}$, $s_{v,t+1}$, $e_{v,t+1}$, $i_{v,t+1}$, $r_{v,t+1}$, $\forall v \in V$). $\log(P(X_{t+1}|X_t))$ can be expressed as in Equation 5.29 where the likelihoods are defined as in Equation (5.30)–(5.33):

$$\log(P(X_{t+1}|X_t)) = \log(\mathcal{L}_s^{t+1}) \ + \log(\mathcal{L}_e^{t+1}) \ + \log(\mathcal{L}_i^{t+1}) \ + \log(\mathcal{L}_r^{t+1}) \tag{5.29}$$

$$\mathcal{L}_{exo} = \prod_{u \in P(v)} (1 - p_{uv})^{i_{u,t} s_{v,t}}$$

$$\mathcal{L}_e^{t+1} = \prod_{v \in V} ((1 - e2i_v)^{e_{v,t} e_{v,t+1}} \ (1.0 - \mathcal{L}_{exo})^{e_{v,t+1}}) \tag{5.30}$$

$$\mathcal{L}_s^{t+1} = \prod_{v \in V} (\mathcal{L}_{exo}^{s_{v,t+1}} (r2s_v)^{r_{v,t} s_{v,t+1}}) \tag{5.31}$$

$$\mathcal{L}_r^{t+1} = \prod_{v \in V} \left((i2r_v)^{i_{v,t} r_{v,t+1}} (1.0 - r2s_v)^{r_{v,t} r_{v,t+1}} \right) \tag{5.32}$$

$$\mathcal{L}_i^{t+1} = \prod_{v \in V} \left((e2i_v)^{e_{v,t} i_{v,t+1}} (1.0 - i2r_v)^{i_{v,t} i_{v,t+1}} \right) \tag{5.33}$$

Each term in $\log(P(X_{t+1}|X_t))$ is additive and log-likelihood terms of endogenous transitions are submodular since they are quadratic terms with negative coefficient. Log-likelihood terms of exogenous transitions are also submodular by following the submodularity proof of the higher-order terms from Theorem 5.4.1.

□

Expressing the inter- and intra-consistency constraints

The inter-consistency constraints (5.23) ensure the validity of diffusion, and they are explicitly written, using binary state variables $s_{v,t}$, $e_{v,t}$, $i_{v,t}$, $r_{v,t}$ for every $v \in V$ and $t \in j, \ldots, k$, as:

$$s_{v,t} + i_{v,t+1} + r_{v,t+1} \leq 1, \tag{5.34}$$

$$e_{v,t} + s_{v,t+1} + r_{v,t+1} \leq 1, \tag{5.35}$$

$$i_{v,t} + s_{v,t+1} + e_{v,t+1} \leq 1, \tag{5.36}$$

$$r_{v,t} + e_{v,t+1} + i_{v,t+1} \leq 1, \ t \in j, \ldots, k-1, \ v \in V \tag{5.37}$$

$$e_{v,t+1} - s_{v,t} \leq \sum_{u \in P(v)} i_{u,t}, \ t \in j, \ldots, k-1, \ v \in V \tag{5.38}$$

Constraints (5.34)–(5.37) ensure that state transitions obey SEIRS dynamics rules such as a node infected at t cannot be susceptible or exposed at $t+1$. The remaining constraint (5.38) ensures that a newly exposed node must have at least one infected predecessor at previous time step. The constraints (5.38) are already represented in the objective function, since $(1.0 - \mathcal{L}_{exo})^{e_{v,t+1}}$ in $\mathcal{L}_e^{t+1}$, and (5.24) takes the lowest possible value $\log(0) = -\infty$ when any of them is not satisfied. So, we can remove the constraints (5.38) without affecting the results. (Some of these constraints are modified accordingly for subset of models. For example, (5.36) becomes $i_{v,t} + s_{v,t+1} \leq 1$ for SI.)

The intra-consistency constraints (5.22) ensure that every node belongs to a single state at each time step:

$$s_{v,t} + e_{v,t} + i_{v,t} + r_{v,t} = 1, \ t \in j+1, \ldots, k-1, \ v \in V \tag{5.39}$$

Optimizing (5.21) **in practice**

Let $E = \{s_{v,t}, e_{v,t}, i_{v,t}, r_{v,t} \,|\, v \in V,\, t \in j, \ldots, k\}$. Then the intra-consistency constraints (5.39) define base of a partition matroid over the ground set E [89], and constraints (5.34)–(5.37) define 2-independence system over the same ground set: Its rank quotient is 2 since the ratio of cardinality of the largest base (maximal independent set) to the cardinality of the smallest base is at most 2. For more detailed information on matroid and independence system, see [60, 124].

Combining (5.24) and (5.21) with the discussion above, the history reconstruction between time steps j and k can be then written as optimizing

$$\max\ T = \sum_{t=j+1}^{k} \log(\mathcal{L}_s^t) + \log(\mathcal{L}_e^t) + \log(\mathcal{L}_i^t) + \log(\mathcal{L}_r^t) \tag{5.40}$$

subject to inter-consistency constraints (5.34)–(5.37) and intra-consistency constraints (5.39). Equation (5.39) cannot be removed as in Section 5.4.1 since each node may belong to any state at time t as we do not know the node states at $t-1$ or $t+1$ (except for boundary times j and k).

When considered together, constraints (5.34)–(5.37) and (5.39) are base of a new matroid defined by the intersection of the partition matroid and 2-independence system. Proof is as follows: Intersection of constraints (5.34)–(5.37) and (5.39) relaxed to $\leq$ define a matroid $\mathcal{M}_p = (E_p, \mathcal{I}_p)$ where $E_p = E$, and independent set $\mathcal{I}_p$ is subset of E satisfying (5.34)–(5.37) and relaxed (5.39). $\mathcal{M}_p$ defines a matroid since all its bases (maximal independent set) have the same cardinality $(k-j-1)|V|$; we can always find a state assignment for every node and every time step that satisfies the constraints, and we cannot assign multiple states to each node at each time step. Then, equality constraints in the original equations (5.39) force independent sets in $\mathcal{I}_p$ to be bases of $\mathcal{M}_p$, as cardinality of an independent set in $\mathcal{I}_p$ will now always be $(k-j-1)|V|$.

This problem becomes non-monotone submodular maximization under matroid base constraints. It is NP-hard [89], and its normalized version can be approximated by $\frac{1}{6}$ by modified local search [89]. We run this method by [89] in *DHR-sub-between* to reconstruct the history between j and k.

DHR-sub-between has three main steps: In the first step, it starts with a base of M_p that also satisfies (5.38), and it finds a base $B_1 \subseteq \mathcal{M}_p$ that is optimal under swap operations. In the second step, it removes B_1 from M_p and greedily finds independent set X_2 that is locally optimal under addition and deletion operations. In the third step, it contracts independent set X_2 from M_p and finds two disjoint bases B_a, B_b that are guaranteed to exist when the original matroid M_p has two disjoint bases. Lastly, it returns the best of three bases B_1, $X_2 \cap \mathcal{B}_a$ or $X_2 \cap \mathcal{B}_b$. The resulting solution always satisfies (5.38) without explicitly checking for them: Local search will not replace the current solution with a low score invalid solution as the objective (5.40) takes the lowest possible value $-\infty$ if any of (5.38) are not satisfied.

5.5. Prize Collecting (Dominating Set) Vertex Cover Relaxations (*DHR-pcdsvc*, *DHR-pcvc*)

Although accurate and practical for smaller graphs, *DHR-sub* may take some time to solve for larger graphs. Then, we can reconstruct the history before the earliest observed snapshot faster

by relaxing *DHR-sub-early*. For a relaxed version of the problem, we define variables differently than above. When reconstructing the history at previous time $j-1$, we define $i_{v,j-1}$, $\forall v \in I_j \cup R_j$, $s_{v,j-1}$, $\forall v \in S_j$, and $e_{v,j-1}$, $\forall v \in E_j$. After this transformation, $\mathcal{L}_e^j$ (5.12) turns into

$$\begin{aligned}
\mathcal{L}_{s2e}^{v,j} &= \prod_{u \in P(v) \cap (I_j \cup R_j)} (1 - p_{uv})^{i_{u,j-1}} \\
\mathcal{L}_{e2e}^{v,j} &= \prod_{v \in E_j} (1 - e2i_v)^{e_{v,j-1}} \\
\mathcal{L}_e^j &= \prod_{v \in E_j} \left(\mathcal{L}_{e2e}^{v,j-1} \ \left(1 - \mathcal{L}_{s2e}^{v,j}\right)^{1 - e_{v,j-1}} \right)
\end{aligned} \tag{5.41}$$

The other likelihoods are transformed similarly.

The hardness of *DHR-sub-early* comes from higher-order terms $\left(1 - \mathcal{L}_{s2e}^{v,j}\right)^{1-e_{v,j-1}}$ in (5.41), so we replace them with their first-order Taylor expansion $\mathcal{T}_{s2e}^{v,j}$ at the point ($i_{u,j-1} = 1$, $\forall u \in P(v) \cap \{I_j \cup R_j\} \cup e_{v,j-1} = 1$) as in (5.42).

$$\mathcal{T}_{s2e}^{v,j} = \log(K_{s2e}) + \frac{1}{K_{s2e}} \sum_{u \in I_j \cup R_j} \frac{\partial \mathcal{L}_{s2e}^{v,j,(I,R)}}{\partial i_{u,j-1}} (i_{u,j-1} - 1) \tag{5.42}$$

In (5.42), $K_{s2e} = \mathcal{L}_{s2e}^{v,j}(1, \ldots, 1) \approx 1$, so the original reconstruction Problem (5.17)–(5.16) for single time step turns into minimizing $-F_r$ as in (5.43)–(5.44):

$$\text{min } -F_r = \sum_{(u,v) \in E^*} w_{uv} \bar{i}_{u,j-1} \, \bar{e}_{v,j-1} + \sum_{u \in I_j \cup R_j} w_u \, i_{u,j-1} + \sum_{v \in E_j} w_v \, e_{v,j-1} + \sum_{v \in S_j} w_v \, s_{v,j-1} \tag{5.43}$$

$$\text{s. t.} \sum_{u \in P(v) \cap \{I_j \cup R_j\}} i_{u,j-1} + e_{v,j-1} \geq 1, \ v \in E_j \tag{5.44}$$

where $\bar{i}_{u,j} = 1 - i_{u,j}$ and $\bar{e}_{v,j} = 1 - e_{v,j}$. The covering constraints (5.44) are inter-consistency constraints ensuring the validity of the diffusion. Similar to *DHR-sub-early*, we do not need intra-consistency constraints since we are reconstructing the history step by step. This problem is *Prize Collecting Dominating Set Vertex Cover* (PCDSVC) over the graph $G^* = (V^*, E^*)$ where $V^* = V$ with weights w_v and directed edge from u to v with weight $w_{uv} = -\log(1 - p_{uv})$ exists when $v \in E_j, u \in P(v) \cap \{I_j \cup R_j\}$ for bipartite $\mathcal{M}$ and $v \in I_j, u \in P(v) \cap \{I_j \cup R_j\}$ for non-bipartite $\mathcal{M}$.

PCDSVC is different than *Vertex Cover* because (1)- We may not cover an edge (u, v) if we pay its price w_{uv}, and (2)- Feasible solution is a vertex dominating set. This problem has not been studied before, it is NP-hard, and it can be approximated by $O(\log(|V^*|))$ by formulating it as *Minimum Hitting Set* and running the greedy method for *Set Cover* as proven in Theorem 5.5.1.

Theorem 5.5.1. *Prize Collecting Dominating Set Vertex Cover (PCDSVC) is NP-hard, and it can be approximated by* $O(\log(|V^*|))$.

Proof. PCDSVC is NP-hard since its special case *Dominating Set* is NP-hard that is obtained when all edge weights are 0 ($w_{uv} = 0$).

Given PCDSVC problem over graph $G^* = (V^*, E^*)$, we construct *Minimum Hitting Set* instance (S, C) as follows: We define the set of elements as $S = \{v \in V^*\} \cup \{e \in E^*\}$ where the cost of each each item in E^* is w_u for every $u \in V^*$ and w_{uv} for every $(u, v) \in E^*$. Subsets $C = C_1 \cup C_2$ of S are defined as: $C_1 = \{e_u, e_v, e_{uv}\}\ , \forall (u, v) \in E^*$ and $C_2 = \{e_u, u \in P(v) \cup \{v\}\}\ , \forall v \in V^*$. This reduction is linear time, approximation preserving and the solution of this *Minimum Hitting Set* gives us the solution for PCDSVC. Here $|S| = |E^*| + |V^*|$ and **Greedy** method for *Set Cover* approximates this problem by $\log(|S|) + 1 \approx O(\log(|E^*| + |V^*|)) + 1 \approx O(\log(|V^*|)) + 1$.

One can also easily show that each *Minimum Hitting Set* instance can be reduced to PCDSVC and this reduction is also approximation preserving. Then, *Minimum Hitting Set* and PCDSVC are *equivalent* under linear reduction and this approximation ratio for PCDSVC is the best we can achieve unless P=NP [45].

□

We can relax this problem further by removing (5.44) and it becomes *Prize Collecting Vertex Cover* (PCVC). PCVC can be approximated by a factor of 2 using the LP relaxation [64], and it can be solved optimally for *bipartite diffusion models* by expressing it as s-t mincut as proven in Theorem 5.5.2.

Theorem 5.5.2. *The Taylor expansion relaxation of Equation 5.17 for bipartite diffusion models can be expressed as s-t mincut problem.*

Proof. The minimization problem for bipartite $\mathcal{M}$ has objective F_{bi} as seen in Equation 5.45. F_{bi} is a regular function [78]: when expressed as the summation of first and second-order terms as in Equation 5.46, each second order term $E^{u,v}(s_{v,j-1}, i_{u,j-1})$ satisfies $E^{u,v}(0,0) + E^{u,v}(1,1) \leq E^{u,v}(0,1) + E^{u,v}(1,0)$ in regular functions. Regular functions can be solved optimally by transforming it into s-t mincut [78]. Transformation is as follows:

$$\min\ -F_{bi} = \sum_{(u,v)\in E^*} \frac{1}{\log(1-p_{uv})}(1 - i_{u,j-1})s_{v,j-1} + \sum_{v \in E_j \cup S_j} w_v s_{v,j-1} + \sum_{v \in I_j \cup R_j} w_v i_{v,j-1} \tag{5.45}$$

$$-F_{bi} = \sum_{u \in I_j \cup R_j, v \in E_j \cup S_j} E^{u,v}(i_{u,j-1}, s_{v,j-1}) \quad + \sum_{v \in I_j \cup R_j} E^v(i_{v,j-1}) + \sum_{v \in S_j \cup E_j} E^v(s_{v,j-1}) \tag{5.46}$$

We define new directed graph $G' = (V', E')$ where $V' = V^* \cup \{s\} \cup \{t\}$. For every $v \in V^*$, we add edge (s, v) with weight $E^v(1)$ if $E^v(1) > 0$ and add edge (v, t) with weight $-E^v(1)$ if $-E^v(1) < 0$. For every $u \in I_j \cup R_j$ and $v \in S_j \cup E_j$, we add edge (u, v) with weight $E^{u,v}(0, 1)$. The s-t mincut solution of this graph gives us the resulting node partition; after the cut edges removed, variables of the nodes that are reachable from s are assigned 1 and the variables of the nodes that have a path to t are assigned 0.

□

The algorithms for these relaxed versions, *DHR-pcdsvc* and *DHR-pcvc*, are similar to *DHR-sub* except they run PCDSVC and PCVC respectively instead of submodular maximization for each iteration.

5.6. Ensemble Initial Spreader Identification

We define *DHR-sub*-ens, *DHR-pcdsvc*-ens and *DHR-pcvc*-ens for the ensemble versions of our methods: they estimate the most likely subset of nodes that explains the diffusion data from multiple runs. For each initial time point seen in the multiple runs, we greedily select the subset of nodes seen in that time point that best explains D in terms of minimum absolute difference F_{dif} from Equation 5.47:

$$F_{dif} = |S_t^e - S_t^t| + |E_t^e - E_t^t| + |I_t^e - I_t^t| + |R_t^e - R_t^t| \tag{5.47}$$

where $S_t^e, E_t^e, I_t^e, R_t^e$ are the set of estimated nodes whereas $S_t^t, E_t^t, I_t^t, R_t^t$ are the set of true nodes for *S*, *E*, *I*, *R* states at time t respectively. We keep adding the node that improves F_{dif} the most until there is no improvement. Lastly, we return set of nodes that has the minimum score among the all possible initial time points as our initial spreader prediction.

5.7. Experimental Results

5.7.1 Comparison and Evaluation

We compared our methods with *NetSleuth*, *Keffectors* and *Rumor* in identifying the initial spreaders. *Keffectors* and *Rumor* require estimates of the number of initial spreaders, so we provide them an estimate of the initial spreader count by the number of clusters in G estimated by modularity [15]. We return the topmost k spreaders from *Rumor* sorted by its rumor centrality metric where k is the number of clusters in G. We also compared our methods with the baseline heuristic *GreedyForward* for history reconstruction that reconstructs the history in each interval by simulating a forward trace starting from the interval's earlier time.

We validated the history reconstruction performance by Kendall Tau-b statistic [2] (τ_B) that measures the similarity between true and estimated node orderings defined in terms of infection times by also adjusting for ties:

$$\tau_B(T, O) = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \tag{5.48}$$

Here, T and O are true and inferred node orderings respectively in terms of given state (such as *infected*). Let V_T be set of nodes seen in true ordering T, then τ_B, n_c are n_d are concordant and discordant pairs respectively, $n_0 = \frac{|V_T|(|V_T|-1)}{2}$, n_1 and n_2 are sum of tied quantities in the true and observed orderings respectively. Kendall tau-b adjusts for ties by subtracting n_1 and n_2 from n_0 in the denominator.

We validated the initial spreaders identification performance by graph-based average matching score ($\overline{M}_G$). Let $\hat{V}_t$ and $\hat{V}_o$ be true and estimated initial nodes respectively, and $G_b = (\hat{V}_t \cup \hat{V}_o, \hat{V}_t \times \hat{V}_o)$ be a weighted bipartite graph with weights $w_{ab} = \frac{1}{1+d_{ab}}$ for every $a \in \hat{V}_t$, $b \in \hat{V}_o$ where d_{ab} is the distance between a and b in G. $\overline{M}_G$ estimates the maximum bipartite matching score in G_b, and returns the average. When $|\hat{V}_o| \neq |\hat{V}_t|$, M_G is modified to account for the unmatched vertices by matching them independently to the best ones.

Both τ_B and $\overline{M}_G$ are normalized, and higher score means better performance in both. We implemented all our methods, synthetic trace generator and existing methods *NetSleuth*, *Rumor*, *Keffectors* in Python, solved LP relaxations by CPLEX [145], and modified and used C++ maximum flow code from [17]. We run all our experiments on Macbook Pro with 2.5 Ghz CPU and 8 GB memory. All our code and data are available on the web[1].

5.7.2 Reconstruction Performance on Synthetic Data

We generated 5 networks of 500 nodes and 5000 edges that are grown by *Erdös-Reyni* [42], *Forest Fire* [90], *linear preferential attachment* [10] network growth models. We generated each synthetic trace by choosing the given number of source nodes randomly, making them infected and running the diffusion over the network until either all nodes become recovered (or infected under the SI model) or until the spread dies out. When multiple snapshots are given, we sample them uniformly in the range (t_{min}, t_{max}).

We test our methods on SI, SIR, SEIR by modeling the transition distributions from a geometric distribution with different parameters in each model in order to assess performance under various conditions. In SI, we selected p_{uv} for every $(u, v) \in E$ uniformly between 0.1 and 0.4. In SIR, we selected $p_{uv}, \forall (u, v) \in E$ uniformly in the range $(0.2, 0.6)$ and $i2r_v, \forall v \in V$ uniformly in the range $(0.5, 0.6)$. In SEIR, we selected $p_{uv}, \forall (u, v) \in E$, $e2i_v, \forall v \in V$, $i2r_v, \forall v \in V$ each uniformly in the range $(0.4, 0.8)$.

DHR-sub and its ensemble version *DHR-sub*-ens perform the best on all the models in terms of identifying the initial spreaders as in Table 5.2. In Table 5.2, dashes represent the methods that cannot be used to reconstruct the diffusion histories, but can only identify the initial spreaders. Its relaxations *DHR-pcdsvc* and *DHR-pcvc* also perform better than the existing methods, and they are good alternatives to *DHR-sub* considering their faster running times. The performance difference between our methods and the existing methods become more apparent especially for SIR and SEIR models.

In terms of history reconstruction, all our methods perform much better than the greedy baseline *GreedyForward*. All our methods perform better when multiple snapshots are available as seen in Figure 5.3 for *DHR-sub* for both SI and SIR. *DHR-sub* reconstructs the histories more precisely when the interval to be reconstructed has lower maximum snapshot ratio ($f_{max} = t_{max}/l_D$) where l_D is the diffusion length, and its performance is not significantly affected by the number of initial spreaders given the same number of snapshots as in Figure 5.3. Lower reconstruction performance for higher f_{max} intervals is due to increasing number of similar quality diffusion histories. In its extreme, τ_B may become close to 0 when reconstructing histories of longer intervals from a single snapshot.

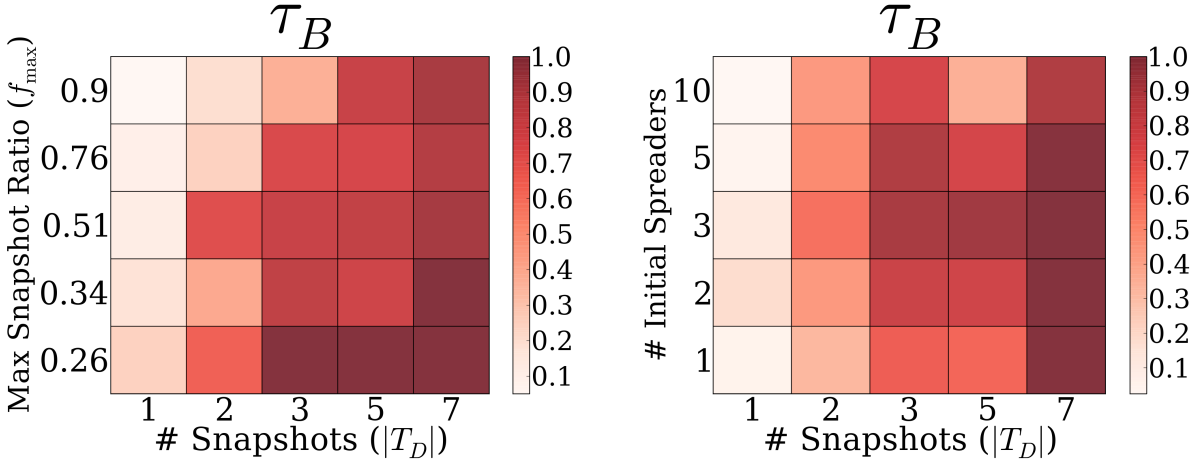
5.7.3 Reconstructing Meme Diffusion History From Blog Data

We used our methods to extract the diffusion history of memes that are defined as short textual phrases that travel through the Web. We inferred the diffusion progression of several memes in two blog networks under SI using the true diffusion data from [58]: *Top-Blog* has 5000 nodes

[1] `http://www.cs.cmu.edu/~ckingsf/software/dhrec`

	Intial Spreader			History			
	FF			*LPA*		*RDS*	
	SI	SIR	SEIR	SI	SIR	SI	SIR
DHR-sub	0.8	0.83	0.81	0.97	0.88	0.69	0.77
DHR-sub-ens	0.87	0.88	0.89	-	-	-	-
DHR-pcdsvc	0.78	0.8	0.81	0.9	0.82	0.64	0.73
DHR-pcvc	0.76	0.76	0.79	0.88	0.77	0.59	0.72
Rumor	0.74	0.7	0.6	-	-	-	-
NetSleuth	0.75	0.8	0.64	-	-	-	-
Keffectors	0.77	0.74	0.7	-	-	-	-
GreedyForward	-	-	-	0.34	0.28	0.31	0.23

Table 5.2: $\overline{M}_G$, τ_B vs. growth and diffusion models for spreader identification (5 true spreaders) and history reconstruction from $|T_D| = 2$ snapshots. Dashes represent the methods that cannot be used to reconstruct the diffusion histories, but can only identify the initial spreaders.



a) SI b) SIR

Figure 5.3: τ_B vs. number of snapshots (x axis) and max snapshot ratio (y-axis), number of true initial spreaders (y-axis) for history reconstruction over *Forest Fire* for *DHR-sub* a) SI b) SIR

and 30072 edges and it shows the connection between the topmost 5000 blogs, *Rand-Blog* has 250 nodes and 3342 edges, and it shows the connection between random 250 blogs. In both networks, nodes represent blogs (personal blogs and mass media), and edges represent hyperlinks from one blog to the another one. We do not know the true p_{uv}, so we estimate them by a geometric distribution with p being 0.3 between mass media, 0.25 from mass media to bloggers, 0.15 between bloggers, and 0.05 from bloggers to media. Traces for several blog topics were obtained from the same source [58]. When tracking the diffusion of a topic, if a blog publishes about it at multiple time points, we assume blog is infected at the earliest time point.

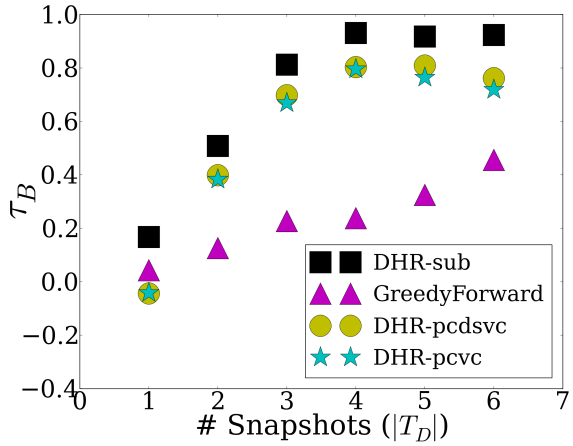
We reconstructed the diffusion history of the memes *Fukushima*, *Arab Spring* and *Nba* on both *Top-Blog* and *Rand-blog* as in Figures (5.4)–(5.5). Values inside the parentheses define the time scale for the meme progression ($1 : 5 = 1$ time unit for 5 days). τ_B are lower than the synthetic case especially when fewer than 2 snapshots are available but they are still reasonable since the true diffusion parameters are unknown. In Figure 5.4, *DHR-sub* performs the best, and all methods reconstruct the diffusion history better when more diffusion data is available. When run with multiple snapshots, *DHR-sub* better captures the diffusion direction and performs almost close to 1 whereas heuristic method *GreedyForward*'s τ_B never exceeeds 0.5. Although *Fukushima* and *Arab Spring* have different diffusion dynamics [58], both trajectories can be reconstructed precisely by *DHR-sub*. Similar to the synthetic case, performance of *DHR-sub* increases if more snapshots are available, and it decreases as f_{max} increases as in Figure 5.5. Overall, both *DHR-sub* and *DHR-pcdsvc* can nicely fill in the missing gaps of the meme diffusion history.

In another example, the order of diffusion estimated by *DHR-sub* matches the true order of the meme *Occupy* reasonably well ($\tau_B = 0.77$). Figure 5.7 shows the true and *DHR-sub* predicted diffusion trajectories of *Occupy* over 50 media sites where red nodes are mass media whereas white ones are personal blogs. Edges between nodes are possible diffusion progression paths. In this case, most of the initial diffusion of *Occupy* happens between mass media, and diffusion at personal blogs start to show up later. However, the speed of the predicted diffusion trajectory is more uniform than the true *Occupy* trajectory.

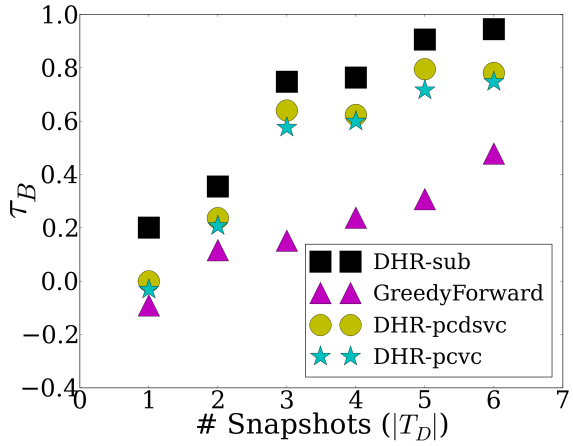
5.7.4 Identifying Initial Water Contamination Sites

We inferred the initial contaminant locations over two water distribution networks [111] where nodes are water demand-supply locations, and the edges represent the water pipes: *Water-sm* has 130 nodes and 173 edges, *Water-big* has 12527 nodes and 14595 edges. We used contaminant diffusion data generated by the water distribution simulator EPANET [118].

We identified the initial contamination sites in *Water-sm* and *Water-big* under SIR where the *recovered* state models the dilution of the contaminant. We approximate the true hydraulic water diffusion dynamics by SIR as follows: we assume that $p_{uv} = K_1/l_{uv}$ and $i2r_{uv} = K_2/l_{uv}$ where K_1, K_2 are constants, and l_{uv} is the length of pipe (u, v). Ensemble methods perform the best as in Figure 5.8 on *Water-sm*, and *DHR-sub* (without ensemble) also performs better than the existing methods. Our methods' performance is consistent across different numbers of initial contamination sites whereas the existing methods' performance is affected by the number of initial sites. Our methods are nonparametric as they do not require number of initial spreaders as input, and our methods' performance consistency makes them the topmost candidates for



a) *Fukushima*



b) *Arab Spring*

Figure 5.4: τ_B vs. number of snapshots for a) *Fukushima*$(1:5)$, b) *Arab Spring*$(1:5)$ on *Top-Blog*

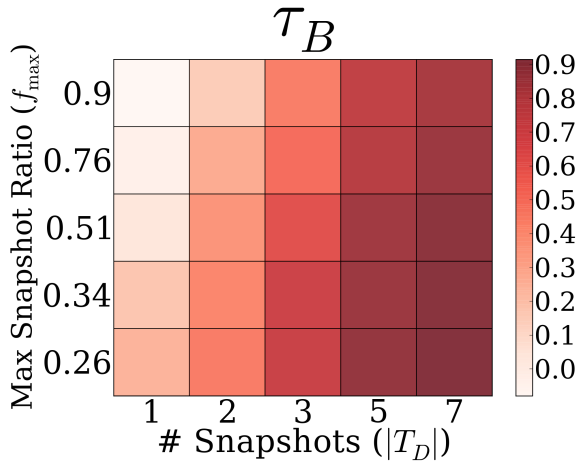


Figure 5.5: τ_B vs. $|T_D|$ and f_{max} for *DHR-sub* of *Nba*$(1:10)$ on *Rand-Blog*

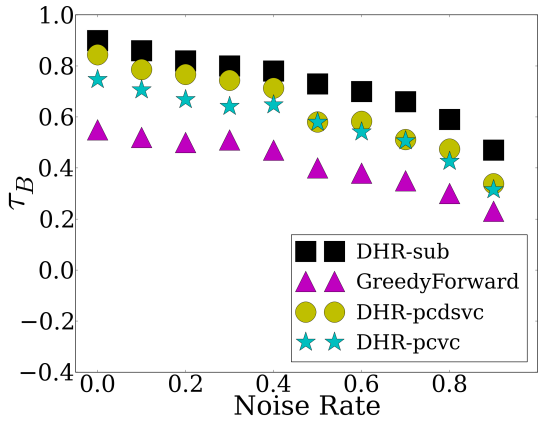
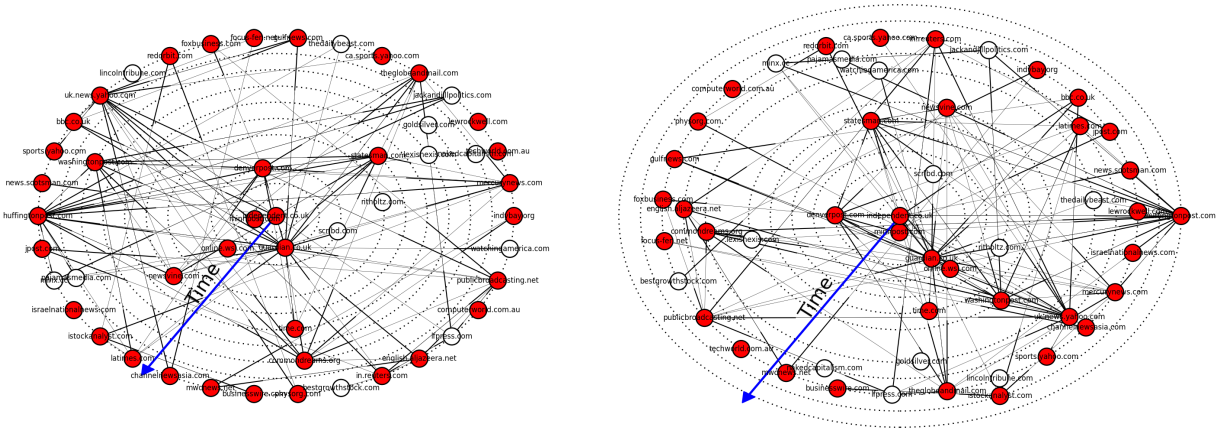


Figure 5.6: τ_B vs. noise ratio (p) over *Water-sm*



a) True Progression of *Occupy* b) *DHR-sub* Predicted Progression of *Occupy*

Figure 5.7: a) True and b) *DHR-sub* predicted diffusion trajectory of *Occupy* over 50 media sites (Red nodes are mass media whereas white ones are personal blogs). Edges between nodes are possible diffusion progression paths.

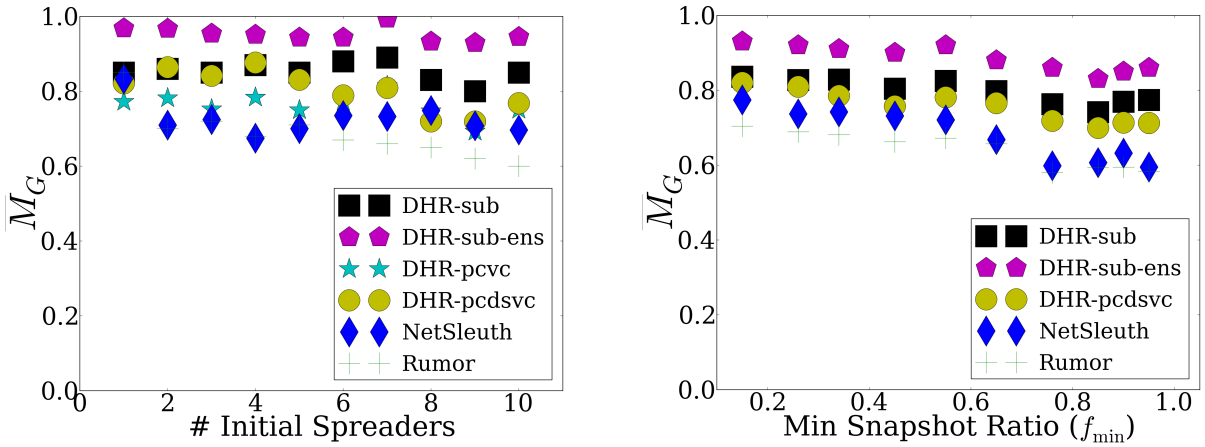
application domains with multiple but unknown number of initial spreaders.

Performance of both *DHR-sub*-ens and *DHR-pcdsvc* decreases for higher f_{min} as in Figure 5.8 on *Water-big*, but they still perform at least 10% better than the best performing *NetSleuth*. This lower performance is due to both difficulty of differentiating between the initial spreader candidates with similar scores, and the decreasing ability to estimate the correct number of initial spreaders. Our methods may miss the true initial spreaders, but their estimates are within close distance to the original spreaders as reflected by higher performance in various cases.

5.7.5 Predicting temporal diffusion features

We may answer questions related to temporal diffusion features from the reconstructed histories such as How quickly did it spread over time?, Did it spread faster at the beginning slowing down at later time steps?, etc. Here, we compared the speed (first-order) and acceleration (second-order) dynamics of *Unemployment* and *Fukushima* estimated from *DHR-sub* with the true ones from [58] as in Figures (5.9)–(5.10). We define speed of a meme as the number of blogs that publishes about the meme for the first time per time unit, and acceleration as the diffusion speed change per time unit.

Unemployment is a more commonly-used meme than *Fukushima*, and such difference is reflected in their diffusion dynamics: *Unemployment*'s diffusion speed is more uniform over time whereas *Fukuhisma* shows more bursty dynamics. Diffusion speed of *Unemployment* has multiple local optima for the time points it peaks in news cycle whereas the speed of *Fukushima* has a single peak when it takes attention of the main media sites. However, such difference in diffusion dynamics does not make a difficulty for *DHR-sub* as *DHR-sub* predicted speed of both memes closely approximate their true ones even from 3 snapshots.



a) *Water-sm* b) *Water-big*

Figure 5.8: a) $\overline{M}_G$ vs. number of initial spreaders for *Water-sm*, b) $\overline{M}_G$ vs. f_{min} for *Water-big* (5 initial sites)

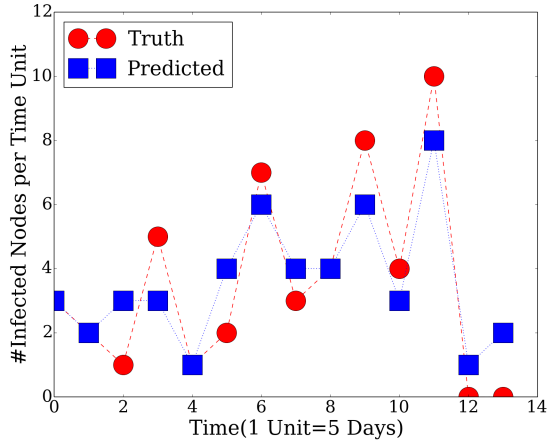
DHR-sub reconstructed histories of both memes also mimick closely their true acceleration dynamics. The change of diffusion speed for *Unemployment* is more uniform than the one for *Fukushima*, and its uniform dynamics are predicted almost perfectly by *DHR-sub* whereas the main peak of *Fukushima*'s acceleration dynamics was missed by *DHR-sub* except precise approximation at remaining time points. Overall, *DHR-sub* reconstructed histories from only 3 snapshots mimick the true speed and acceleration dynamics of both memes quite precisely even though the original prediction scores are below 0.75 ($\tau_B = 0.74$ for *Unemployment*, $\tau_B = 0.65$ for *Fukushima*).

5.7.6 Scalability and Robustness of History Reconstruction

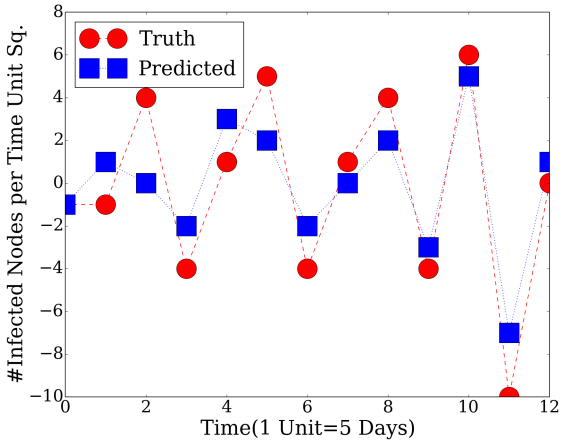
All our methods reconstruct the history on *Top-Blog* in less than 2 minutes, and our relaxation methods *DHR-pcdsvc*, *DHR-pcvc* reconstruct the history in less than 10 minutes on a large 2D grid graph having 90000 nodes and 179400 edges, with reasonable performance ($\tau_B = 0.71, 0.63$) as in Table 5.3 whereas *DHR-sub* takes more than an hour on a personal laptop. When combined with previous sections' results, running times in Table 5.3 suggest that *DHR-pcdsvc* and *DHR-pcvc* are nice alternatives to *DHR-sub* for scalable history reconstruction on large graphs. However, we still need faster methods for scalable reconstruction on million-node graphs.

Figure 5.6 shows the performance of contaminant diffusion history reconstruction over *Water-sm* under increasing noise levels. Let p be the noise ratio between 0.0 and 1.0, and we added the synthetic noise p as follows: For each node and each state, we randomly select a value m between 0 and pl_D where l_D is length of the diffusion and flip a coin to either add m to the current state transition time t_v, or subtract it from t_v. If modified transition time $(t_v + m)$ is less than 0, we make it 0.

Our methods do not show a sudden performance drop by increasing noise levels, as *DHR-sub* can still reconstruct histories with performance over $\tau_B = 0.7$ even when the noise levels are 0.5.

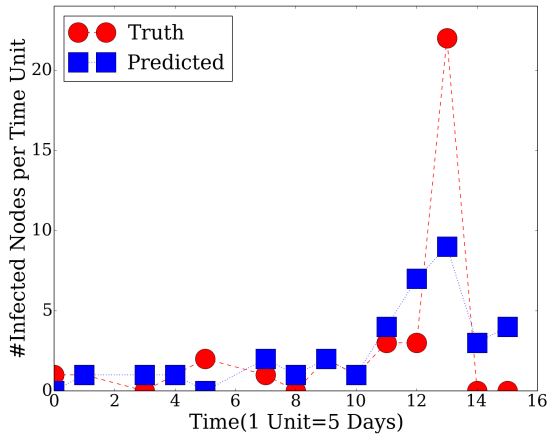


a) Speed

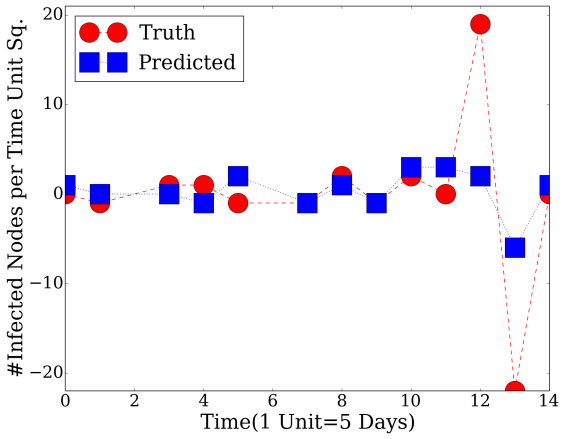


b) Acceleration

Figure 5.9: a) Speed, b) Acceleration dynamics of true and predicted diffusion of *Unemployment* over time from 3 snapshots



a) Speed



b) Acceleration

Figure 5.10: a) Speed, b) Acceleration dynamics of true and predicted diffusion of *Fukushima* over time from 3 snapshots

	Top-Blog		*2D-GRID*	
	$\lvert T_D\rvert = 1$	$\lvert T_D\rvert = 3$	$\lvert T_D\rvert = 1$	$\lvert T_D\rvert = 3$
DHR-sub	112.5	48.9	-	-
DHR-pcdsvc	53.9	28.2	592.1	199.2
DHR-pcvc	49.9	14.3	351.7	82.7

Table 5.3: History reconstruction time (in seconds) for *Top-Blog* and a 2D grid graph for different numbers of diffusion snapshots.

Similarly, *DHR-sub*-ens achieves $\overline{M}_G = 0.72$ in identifying the initial contaminant locations over *Water-sm* for $p = 0.5$ (results are not shown). In general, all our methods are robust to the noise in the diffusion data.

5.8. Conclusions

We designed several methods for estimating diffusion histories that either optimize the likelihood or its relaxations with provable performance guarantees for local steps. Our methods do not require the number of initial spreaders and diffusion length as parameters. They identify the initial spreaders better than the existing methods specially designed for this task. They reconstruct the history accurately in a number of scenarios. We also accurately estimated temporal diffusion characteristics of several semantically different memes from partial data. These findings suggest the reconstructability of diffusion history from partial data under several settings. Partial diffusion data is not an unsolvable bottleneck as missing diffusion history can be completed by our methods accurately.

Chapter 6

Deconvolution Of Ensemble Chromatin Interaction Data Reveals The Latent Mixing Structures In Cell Subpopulations

A preliminary version of this chapter appeared in the 19th International Conference on Research in Computational Molecular Biology, RECOMB 2015 with the title *Deconvolution Of Ensemble Chromatin Interaction Data Reveals The Latent Mixing Structures In Cell Subpopulations* [129].

6.1. Introduction

The spatial organization of the genome as it is packed into the cell is closely linked to its function. Chromatin loops as well as locally clustered topological domains [35] play a role in long-range transcriptional regulation [6, 59, 122] and the progression of cancer [51, 158]. For instance, the impact of the long-range interacting gene clusters in the conformation of HOXA cluster is better understood in the context of the genome's three-dimensional relationships [120]. Expression in the beta-globin locus is mediated by folding to bring an enhancer and associated transcription factors within close proximity of a gene [13, 142]. Loci of mutations that affect expression of genomically far-away genes (eQTLs) are statistically significantly closer in 3D to their regulated genes than expected by a stringent null model [40], indicating that 3D contacts play a widespread role in gene regulation. Measuring and modeling the three-dimensional shape of eukaryotic and prokaryotic genomes is thus essential to obtain a more complete understanding of how genomes function.

A class of recently introduced experimental techniques called chromosome conformation capture (3C) allows for the measurement of pairwise genomic contacts at much higher resolutions than FISH microscopy experiments [33]. These techniques cross-link spatially close fragments of the genome within a population of millions of cells and use high-throughput sequencing to determine which fragments were cross linked together. Since the development of the original 3C method, a number of enhancements to the protocol such as 3C, 4C, 5C, Hi-C, and TCC, have been introduced [39, 71, 94, 137]. Genome-wide interactions from Hi-C experiments, for example, can be analyzed at fragment lengths as low as 10kb [70], though resolutions of 20-

40kb are more common. Here, for simplicity, we refer to all 3C-like techniques as 3C. All of these methods result in a matrix $\mathbf{F} : V \times V \rightarrow \mathbb{R}_0^+$ where $V = \{1, 2, \ldots, n\}$ is the set of genome fragments and where $F_{i,j}$ is the number of times genome fragment i was observed in close proximity to fragment j within the assayed population of cells. Under the assumption that these contact events will be more common for spatially close pairs as shown in [141], the counts can be converted into spatial distances. The count matrix $\mathbf{F}$ or its associated distance matrix are then analyzed in the context of long-range gene regulation or used to produce three dimensional models of the genome [146].

A challenge with 3C data is that it is collected over a population of cells. The genome structures within these cells vary since (1) They exist at different points in time within a particular phase of the cell cycle, (2) They may be associated with different methylation and therefore heterochromatin formations [11], and (3) Chromatin itself can fluidly take on different three-dimensional forms. Analysis of the combined matrix $\mathbf{F}$ therefore may be misleading.

We tackle the problem of extracting the genome contact map of each subpopulation of cells from the combined, ensemble matrix $\mathbf{F}$. A subpopulation represents cells with similar interaction matrices and can model cells in distinct subphases in the cell cycle (e.g. early G1 vs. late G1), cells that are undergoing different gene expression programs, or cells that are in different stochastic structural states. We present a method to deconvolve the observed $\mathbf{F}$ into a collection of biologically-plausible, unobserved subpopulation matrices $\mathbf{F^i}$ such that

$$\mathbf{F} \approx \sum_i \lambda_i \mathbf{F^i}, \tag{6.1}$$

where λ_i are the relative abundances (densities) of cells in each subpopulation (class) i. This is the *3C Deconvolution Problem* (*3CDE*), which we show to be NP-hard whether λ_i are in $\mathbb{R}$ or $\mathbb{N}$.

To solve this problem, we assume that the interaction matrix $\mathbf{F^i}$ of each class is composed of *nonoverlapping* topological domains that are highly self-interacting consecutive genomic intervals. Such topological domains have been widely observed and are a natural unit of genome structure [14, 35]. We model these domains here using a particular type of quasi-clique, allowing for missing interactions within a densely interacting domain. The algorithm supports the use of prior knowledge of topological domain structure as estimated from the ensemble matrix $\mathbf{F}$ or through other means that inform the choice of domains that appear in each $\mathbf{F^i}$. We explore two variants of our algorithm: one called *3CDEint* in which the class densities λ_i are required to be integers and one called *3CDEfrac* in which they are not. The integer case is appropriate when the matrix $\mathbf{F}$ contains unnormalized counts, while the real-valued version is appropriate when $\mathbf{F}$ has been normalized to account for experiment bias [156].

Both *3CDEint* and *3CDEfrac* solve *3CDE* in an iterative two-step fashion that alternates between optimizing the matrices $\mathbf{F^i}$ (Step 1 in Sec. 6.2.3) and then optimizing the densities λ_i (Step 2 in Sec. 6.2.4). We show that each step can be solved near-optimally. These two steps use non-monotone supermodular optimization and SDP relaxations, respectively. For smaller problem instances, we develop optimal methods *3CDEint*-opt and *3CDEfrac*-opt based on Quadratic Integer Programming that allow us to compare our approximate solutions of *3CDEint* and *3CDEfrac* to the true optimal solutions.

We show that our estimated deconvoluted matrices and topological domain structures are very similar to those derived from ground truth single cell data [104] as well domain structures

in particular cell phases [105]. We also show that domain boundaries from deconvolved matrices are often more enriched or depleted for regulatory chromatin markers H3K4me3, H3K36me3, H3K9me3 and CTCF when compared to boundaries from convolved matrices. The deconvolved domain substructures we produce may therefore be more useful in analyses of long-range regulation with respect to chromatin structure, and our methods can be used as way to simultaneously find domains while determining population substructures.

6.1.1 Related Work

Most existing methods for finding domains within 3C matrices [35, 50] and for embedding 3C matrices in 3D space [94, 159] treat 3C interaction data as a single unit ignoring the fact that it is an ensemble over millions of cells. Although none of the existing methods explicitly solve the deconvolution problem, some [50, 65, 71, 119] find multiple 3D embeddings or multiple domain decompositions. For example, Rousseau et al. (2001) [119] develop an MCMC sampling technique *MCMC5C*, and Hu et al. (2013) [65] develop *BACHMIX* that optimizes likelihood over a mixture model to find multiple embeddings. Neither of these methods considers the additive affects of interactions. Another method discussed in Kalhor et al. [71] generates a population of structures by restricting the number of times each interaction is involved in a solution, which may mimic the deconvolution to a certain extent but ignores the domain structure of the genome. *Armatus* [50] finds multiple optimal and near-optimal domain decompositions at multiple scales by optimizing a density-like objective. None of these methods determine domain substructures or population densities of these substructures.

On the experimental side, two recent Hi-C modifications try to limit the effect of cell-to-cell variations. Nagano et al. (2013) [104] carry out experiments on single cells that come at a higher experimental cost and produce lower-resolution interaction matrices. Another modification measures the interactions at a particular cell phase by arresting the population of cells at that phase by thymidine and nocodazole. However, these chemicals may disrupt the original genome structure [87, 105]. Since single cell 3C data [104] is so recent, we provide the first comparison of deconvoluted structures to real single cell matrices.

6.1.2 The Deconvolution Problem (*3CDE*)

We want to estimate the interaction matrices $\mathbf{F^i}$ of the subpopulations. Without additional constraints, deconvolution is under-constrained because an infinite number of matrices can explain the ensemble data equally well. However, we can exploit the fact that a 3C interaction matrix is (1) fairly dense around the diagonal due to the abundance of short-range interactions even being sparse overall, and (2) composed of topological domains that are highly self-interacting, non-overlapping genomic intervals that are the building blocks of genome [14, 35].

We encode these assumptions by modeling topological domains as *bandwidth-quasi-cliques* (*BQC*s) to allow domain structures to be locally dense while not requiring all interactions to exist. A d-*BQC* is a defined by a genomic subrange $[s_p, e_p]$ where there is an interaction between every pair of fragments that are separated by at most d fragments, resulting in a banded pattern of interactions. Figure 6.1 shows a *BQC* for a 6-loci domain at 1 mb resolution. Let $l_{\min}$ and $l_{\max}$ be minimum and maximum possible domain sizes ($l_{\min} \leq e_p - s_p + 1 \leq l_{\max}$). There

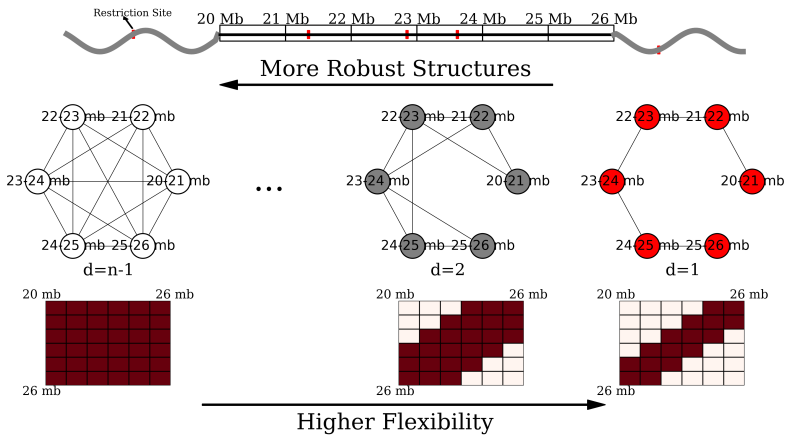


Figure 6.1: *d-bandwidth-quasi-clique* (*d-BQC*).

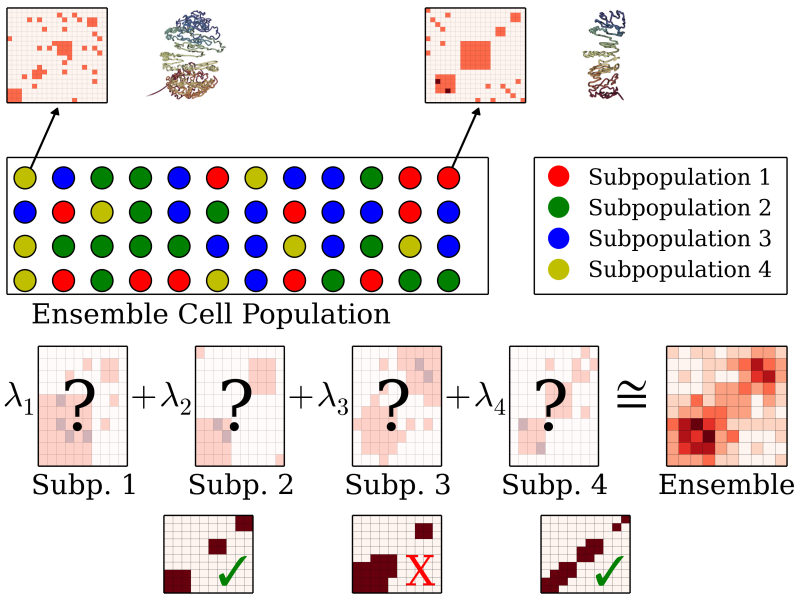


Figure 6.2: *3CDE*: Given the ensemble matrix, we infer the mixing matrices in terms of *BQC*s and the densities λ's without letting *BQC*s overlap in each subpopulation.

are $e_p - s_p$ possible *BQC*s for a domain p covering the range $[s_p, e_p]$, so total number of *BQC*s over all domains is $\sum_{l=l_{\min}}^{l_{\max}} (n - l + 1)(l - 1) = O\left(n(l_{\max} - l_{\min})^2\right)$, where n is the number of fragments.

We assume that the observed ensemble matrix $\mathbf{F}$ is sum of binary interaction matrices ($\{\mathbf{F^1}, \mathbf{F^2}, \ldots, \mathbf{F^k}\}$), each multiplied by their densities ($\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$). We further assume that each $\mathbf{F^i}$ is composed of non-overlapping *BQC*s. Finally, we assume that the number of classes k is given or it can be easily estimated as each subpopulation is a meaningful distinct unit such as different phase of the cell cycle. Let $I = \{1, \ldots, k\}$ be the set of class labels. Figure 6.2 illustrates *3CDE*, which is defined formally below:

Problem 3 (*3CDE*)**.** *We are given an ensemble interaction matrix* $\mathbf{F}$*, a number of classes* k*, and (optionally) a set of prior domains* P_c*. For each class* i*, we want to choose a set of nonoverlapping bandwidth-quasi-cliques and density* λ_i *such that the squared Frobenius norm of the difference between* $\mathbf{F}$ *and the sum of the matrices* $\mathbf{F^i}$ *derived from the chosen bandwidth-quasi-cliques is minimized.*

6.2. Approximate 3C Deconvolution Methods

6.2.1 Mathematical Formulation and Hardness

We formulate the *3CDE* problem using a three-part objective that (1) minimizes squared Frobenius norm of the difference between observed convolved matrix and convolution of the deconvolved matrices, (2) maximizes the quality of domains defined by their *BQC*s, and (3) maximizes the overlap with a prior set of candidate domains P_c if available. Formally, given minimum and maximum domain sizes $l_{\min}$ and $l_{\max}$, let $P = \{[s_p, e_p] \mid s_p \in 1, \ldots, n,\, e_p \in s_p + l_{\min} - 1, \ldots, \min(n, s_p + l_{\max} - 1)\}$ be the set of possible domains, and $M : V \to 2^P$ be a function that maps each 3C fragment to the set of domains to which it could belong:

$$M(v) = \{p \mid \forall p = [s_p, e_p] \in P,\, s_p \leq v \leq e_p\}$$

Define $G_q = (V_q, E_q)$ to be the *BQC* intersection graph where

$$V_q = \{(p, d) \mid p \in P,\, d \in 1, \ldots, l_p - 1\} \qquad \text{(the set of possible } \textit{BQC}\text{s)} \tag{6.2}$$

$$E_q = \{((p_i, d), (p_j, t)) \mid (p_i, d), (p_j, t) \in V_q^2,\, i \neq j,\, p_i \cap p_j \neq \emptyset\} \tag{6.3}$$

A pair (p, d) represents a ***BQC*** by its domain and bandwidth d and l_p is the number of fragments in domain p. We can express *3CDE* as:

$$\min \quad \overbrace{\sum_{(u,v)\in V^2} \left(F_{u,v} - \left(\sum_{i\in I} \lambda_i \left(\sum_{p\in M(u)\cap M(v)} \sum_{d=|u-v|}^{l_p-1} x_{pdi} \right) \right) \right)^2}^{\left\| \mathbf{F} - \sum_{i\in I} \lambda_i \mathbf{F^i} \right\|_F^2} + \underbrace{\sum_{i\in I} \sum_{(p,d)\in V_q} w_{pd}(1 - x_{pdi})}_{\text{Domain Weakness}}$$

$$+ \quad \underbrace{\lambda^p \sum_{i\in I} \sum_{p\in P_c} \sum_{d\in 1,\ldots,l_p-1} (1 - x_{pdi})}_{\text{Distance From Prior}} \tag{6.4}$$

$$\text{s.t.} \quad x_{pdi} + x_{rti} \leq 1, \qquad \forall \left((p, d), (r, t)\right) \in E_q, \forall i \in I \tag{6.5}$$

$$x_{pdi} \in \{0, 1\}, \qquad \forall (p, d) \in V_q, \forall i \in I \tag{6.6}$$

where $x_{pdi} = 1$ if d-***BQC*** of interval p is assigned to class i. Here, d ranges from $|u - v|$ to $l_p - 1$ for each entry (u, v) since d-***BQC*** of p can correspond to matrix entries up to d away from the diagonal. Eqns. (6.5) ensures each $\mathbf{F^i}$ is made up of nonoverlapping ***BQC***s. We penalize for selecting less dense (weaker) ***BQC***s where w_{pd} defines the quality of d-***BQC*** of p. We also reward larger overlaps with the prior candidate domains P_c from domain finders, such as *Armatus*, by minimizing the distance from the prior domains where λ^p is weight of the prior.

3CDE has two variants depending on the class densities: (1) *3CDEint* where λ_i are integers, and (2) *3CDEfrac* where λ_i can take any nonnegative values (useful for normalized $\mathbf{F}$). *3CDE* is NP-complete whether λ_i are in $\mathbb{R}$ or $\mathbb{N}$ as proven in Theorem 6.2.1, and *3CDEint* can be solved exactly in pseudo-polynomial $O(kn^{4k-1}F_{max}^k)$ time by dynamic programming. Similarly, it can be approximated with *Set Cover* [147] by using a pseudo-polynomial number of constraints as proven in Theorem 6.2.2. However, this approach is impractical, and prohibitively slow for large n, k, and $F_{max} = \max\{F_{i,j}\}$.

Theorem 6.2.1. 3CDE *is NP-complete.*

Proof. There are two *3CDE* variants depending whether λ are in $\mathbb{R}$ or $\mathbb{N}$. First, we prove that *3CDEint* is NP-complete by proving it is still NP-complete for the special case when class densities $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ are also given. We define this problem *3CDEint*$_\Lambda$, and its decisional variant as *Decisional 3CDEint*$_\Lambda$ as below:

Decisional 3CDEint*$_\Lambda$:** Given F, integer Λ, w_{pd}^c, and a rational number r, determine whether there is an nonoverlapping ***BQCs such that $\left\| \mathbf{F} - \sum_{i\in I} \lambda_i \mathbf{F^i} \right\|_F^2 + \sum_{i\in I} \sum_{(p,d)\in V_q} w_{pd}^c(1-x_{pdi}) \leq r$.

3CDEint is clearly in NP since yes instances can be verified in polynomial time, and its solution is polynomial in the input size. To verify NP-hardness, we reduce *SUBSET SUM* to *Decisional 3CDEint*$_\Lambda$ to prove NP-hardness of *Decisional 3CDEint*$_\Lambda$. This result implies NP-hardness of the optimization variant *3CDEint*$_\Lambda$ since its output can be used to answer *Decisional 3CDEint*$_\Lambda$. Recall that an instance of *SUBSET SUM* problem is given by $k + 1$ integers

$a_1, a_2, \ldots, a_k, S$, and our goal is to decide whether there is a set $A \subseteq \{1, \ldots, k\}$ such that $\sum_{i \in A_i} a_i = S$. Given a *SUBSET SUM* instance, we reduce it to *Decisional 3CDEint*$_\Lambda$ instance $(\mathbf{F}, \Lambda, w^c_{pd}, r)$ as follows:

We form the following parameters for given *SUBSET SUM* instance for reduction:

$$\mathbf{F} = \begin{bmatrix} S & S & \ldots & 0 \\ S & S & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix}_{n \times n} \qquad \Lambda = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix}_{k \times 1} \qquad w^c_{pd} = [0,\ 0,\ \ldots,\ 0]_{1 \times V_q} \qquad r = 0$$

where each $\mathbf{F^i}$ in *3CDEint*$_\Lambda$ is defined as:

$$\mathbf{F^i} = \begin{bmatrix} \overbrace{x_{(1,2),1,i} + x_{(1,3),1,i} + \ldots + x_{(1,n),n-1,i}}^{F^i_{1,1}} & \overbrace{x_{(1,2),1,i} + \ldots + x_{(1,n),1,i}}^{F^i_{1,2}} & \ldots & \overbrace{x_{(1,n),1,i} + \ldots + x_{(1,n),n-1,i}}^{F^i_{1,n}} \\ x_{(1,2),1,i} + x_{(1,3),1,i} + \ldots + x_{(1,n),1,i} & x_{(1,2),1,i} + \ldots + x_{(2,n),n-2,i} & \ldots & x_{(1,n),1,i} + \ldots + x_{(2,n),n-2,i} \\ & \ddots & \ddots & \\ \ldots & & & \ldots \\ x_{(1,n),1,i} + \ldots + x_{(1,n),n-1,i} & x_{(1,n),1,i} + \ldots + x_{(2,n),n-2,i} & \ldots & x_{(1,n),1,i} + \ldots + x_{(n-1,n),1,i} \end{bmatrix}_{n \times n}$$

where $x_{(s,e),d,i} = 1$ if *d-BQC* of domain $[s, e]$ is used in class i. Each entry $F^i_{a,b}$ consists of *BQC* variables that include a and b. Each $F^i_{a,b}$ can be at most 1 due to nonoverlapping *BQC* constraints.

Clearly, this reduction can be done in polynomial time. If $(a_1, a_2, \ldots, a_k, S)$ is a *YES* instance of *SUBSET SUM*, then there is a set $A \subseteq \{1, \ldots, k\}$ such that $\sum_{i \in A_i} a_i = S$. There are two cases: If the solution returned by this *Decisional 3CDEint*$_\Lambda$ is 0, this means a *YES* for *SUBSET SUM* since this is the only way it returns 0, otherwise it would have been greater than 0. Similarly, if the solution is not 0, this is *NO* for *SUBSET SUM* since it would have returned *YES* otherwise. If such solution exists, it can only exist in $\{x_{(1,2),1,1}, x_{(1,2),1,2}, \ldots, x_{(1,2),1,k}\}$ since the rest of variables are also seen outside $F_{0:2,0:2}$, and they must add up to 0 for *YES*. In this case, optimization variant *3CDEint*$_\Lambda$ can be solved in polynomial time given an oracle for the decisional variant.

Similarly, *3CDEfrac*$_\Lambda$ with nonnegative densities is also NP-complete since it is a generalization of *3CDEint*$_\Lambda$. Then, *3CDEfrac* is also NP-complete.

□

Theorem 6.2.2. 3CDEint *can be approximated to a factor of* 3 *via the greedy method for Set Cover.*

Proof. Number of possible values for each λ_i is limited in the objective (6.4) for *3CDEint*; $\lambda_i \in S = \{0, 1, \ldots, F_{max}\}$. In this case, we can reformulate *3CDEint* in terms of single type of variable by combining the assignment and density variables into a single variable: We define a binary variable x_{pdis} for every $(p, d) \in V_q$, $i \in I$, $s \in S$ where $x_{pdis} = 1$ if *d-BQC* of domain p is assigned to class i which has density s. This reformulation will introduce the following new constraints:

- *Single density assignment in each class:* None of *BQC* pairs assigned to the same class can have different densities as satisfied by:

$$x_{pdis_1} + x_{rtis_2} \leq 1, \quad \forall (p, d) \neq (r, t) \in V_q^2,\ \forall i \in I,\ \forall s_1 \neq s_2 \in S^2$$

Then, *3CDEint* can be expressed as *Set Cover* [147] problem by replacing $y_{pdis} = 1 - x_{pdis}$, and replacing each quadratic term $x_{pdis_1} x_{rtjs_2}$ in the objective (6.4) by a single variable $z_{pdis_1,rtjs_2}$ and adding the following constraint:

$$z_{pdis_1,rtjs_2} + y_{pdis_1} + y_{rtjs_2} \geq 1$$

to ensure that $z_{pdis_1,rtjs_2} = 1$ if both $y_{pdis_1} = 0$ and $y_{rtjs_2} = 0$. The resulting program for *Set Cover* has a pseudo-polynomial number of variables and constraints both on the order of $O\left(\frac{k(k-1)}{2}(F_{max}+1)^2 |E_q|\right)$ as expressed in (6.7)–(6.12):

$$\min \sum_{(p,d),(r,t)\in E_q} \sum_{(i\neq j)\in I^2} \sum_{s_1,s_2\in S^2} \overline{c}_{xpis_1,rtjs_2}\, z_{pdis_1,rtjs_2} + \sum_{(p,d)\in V_q} \sum_{i\in I} \sum_{s\in S} \underbrace{\left(c_{pdis} + w^c_{pd}\right)}_{\overline{c}_{xpis}} y_{pdis} \tag{6.7}$$

$$\text{s.t} \quad y_{pdis} + y_{rtis} \geq 1, \quad \forall (p,d),(r,t) \in E_q,\, \forall i \in I,\, \forall s \in S \tag{6.8}$$

$$z_{pdis_1,rtjs_2} + y_{pdis_1} + y_{rtjs_2} \geq 1, \quad \forall (p,d),(r,t) \in E_q,\, \forall i \neq j \in I^2,\, s_1, s_2 \in S^2 \tag{6.9}$$

$$y_{pdis_1} + y_{rtis_2} \geq 1, \quad \forall (p,d) \neq (r,t) \in V_q^2,\, \forall i \in I,\, \forall s_1 \neq s_2 \in S^2 \tag{6.10}$$

$$y_{pdis} \in \{0,1\}, \quad \forall (p,d) \in V_q,\, \forall i \in I,\, \forall s \in S \tag{6.11}$$

$$z_{pdis_1,rtjs_2} \in \{0,1\}, \quad \forall (p,d),(r,t) \in E_q,\, \forall i \neq j \in I^2,\, s_1, s_2 \in S^2 \tag{6.12}$$

where $\overline{c}_{pdis_1,rtjs_2}$ and $\overline{c}_{xpis} = c_{xpis} + w^c_{pd}$ are the coefficients of quadratic and linear terms respectively when (6.4) is expressed as the objective (6.7). This *Set Cover* instance can be approximated by 3 via greedy method [147] since maximum set size is 3. This pseudo-polynomial number of variables and constraints become prohibitively large for realistic *3CDEint* instances.

□

6.2.2 Practical Approximate Methods

Due to hardness of *3CDE*, we design the approximate methods *3CDEfrac* and *3CDEint* for integer and real-valued class densities respectively. Both methods are similar, so we explain *3CDEint* in detail and discuss the differences between *3CDEfrac* from *3CDEint* in the last subsection. Let $S = \{0, 1, \ldots, F_{max}\}$ be the set of integer subpopulation densities where $F_{max} = \max\{F_{i,j}\}$, and we define $y_{is} = 1$ if subpopulation i's density is s. Program (6.4)–(6.6) can be expressed as

constrained minimization of the biset function $Q(X,Y)$ as in Program (6.13)–(6.17):

$$\min Q(X,Y) = \sum_{(u,v)\in V^2} \left(F_{u,v} - \left(\sum_{i\in I} \sum_{s\in S} s y_{is} \Big(\sum_{p\in M(u)\cap M(v)} \sum_{d=|u-v|}^{l_p-1} x_{pdi} \Big) \right) \right)^2 + \sum_{i\in I} \sum_{(p,d)\in V_q} w^c_{pd}(1 - x_{pdi}) \tag{6.13}$$

$$\text{s.t} \quad x_{pdi} + x_{rti} \leq 1, \quad \forall \left((p,d),(r,t)\right) \in E_q, \forall i \in I \tag{6.14}$$

$$\sum_{s\in S} y_{is} = 1, \quad \forall i \in I \tag{6.15}$$

$$x_{pdi} \in \{0,1\}, \quad \forall (p,d) \in V_q, \forall i \in I \tag{6.16}$$

$$y_{is} \in \{0,1\}, \quad \forall i \in I, \forall s \in S \tag{6.17}$$

where $w^c_{pd} = w_{pd} + \lambda^p$ is the combined domain prior and robustness weight. The nonoverlapping ***BQC*** constraints (6.14) depend only on X, and (6.15) ensures a single density assignment for each class. We solve Program (6.13)–(6.17) iteratively in two steps starting with unit class densities. We describe these two steps with their approximation guarantees in detail below. Intuitively, the first step tries to find the best ***BQC*** assignments X given the class densities Y, while the second step tries to find the best Y given X. These steps are iterated until convergence.

6.2.3 Step 1: Non-monotone Supermodular Optimization for Estimating Mixing Matrices

When the class densities Y are given, (6.15) disappears, and the objective is slightly modified as in:

$$\min Q(X|Y) = \sum_{(u,v)\in V^2} \left(F_{u,v} - \left(\sum_{i,s\in Y} s \Big(\sum_{p\in M(u)\cap M(v)} \sum_{d=|u-v|}^{l_p-1} x_{pdi} \Big) \right) \right)^2 + \sum_{i\in I} \sum_{(p,d)\in V_q} w^c_{pd}(1 - x_{pdi}) \tag{6.18}$$

This is *Minimum Non-monotone Supermodular Independent Set in the Interval Graph* defined by the ***BQC*** intersection graph G_q since objective (6.18) is non-monotone supermodular. We solve its fractional relaxation optimally, round the fractional solution via $(1, e^{-1})$-balanced contention resolution scheme by [48] 100 times, and return the minimum solution. This scheme gives $\frac{1}{e} + (1 - \frac{1}{e})\overline{Q}$ approximation guarantee as in Lemma 6.2.3 where $\overline{Q} = \frac{Q(\emptyset,\emptyset)+\epsilon}{k \min_{(p,d)}\left(w^c_{pd}\right)+\epsilon}$ for arbitrarily small constant $\epsilon > 0$. This bound is also preserved up to an additive error for large matrices which weights are usually estimated by sampling [140]. Each rounding step is defined as follows: For each class i, we choose a ***BQC*** with probability $1 - e^{-x_{pdi}}$ to put into the solution R. After sampling, we mark the ***BQC*** represented by x_{pdi} for deletion if there is a different ***BQC*** in R that intersects the starting point of p. After removing all marked ***BQC***s from R, we return independent set R as a solution.

We can achieve similar approximation bound by transforming the program into *Set Cover* where (1) we replace every x_{pdi} with $\hat{x}_{pdi} = 1 - x_{pdi}$, (2) define a variable for each quadratic term, and (3) introduce extra covering constraints to enforce the quadratic costs when none of its linear terms are added. This *Set Cover* can be solved by greedy method which runs faster for large matrices.

Lemma 6.2.3. *Step 1 can be approximated to a factor* $\frac{1}{e} + (1 - \frac{1}{e})\overline{Q}$.

Proof. The problem of maximizing $\hat{Q}(X|Y) = -Q(X|Y)$ is maximum non-monotone submodular independent set in interval graph. We can make $\hat{Q}(X|Y)$ nonnegative submodular function as in $\hat{Q}^n(X|Y) = \hat{Q}(X|Y) + A$ where $A = Q(\emptyset, \emptyset) = \sum_{(u,v)\in V^2} F_{u,v}^2 + \sum_{i\in I} \sum_{(p,d)\in V_q} w_{pd}^c$ is the initial empty solution. Let X^* be the optimal solution, and X^r be the solution returned by $(1, \frac{1}{e})$ monotonic contention resolution scheme of [48] described in Section 6.2.3, nonnegative $\hat{Q}^n(X|Y)$ can be approximated by $\frac{1}{e}$ as in (6.19):

$$\hat{Q}^n(X^r|Y) \geq \hat{Q}^n(X^*|Y)\frac{1}{e} \tag{6.19}$$

Here, $\frac{1}{e}$ ratio is derived as follows: We first optimize the fractional relaxation of the quadratic Program optimally since it is convex. Then, rounding the solution via (b, e^{-b}) monotone contention resolution scheme gives $\alpha b e^{-b}$ approximation ratio where $\alpha = 1$ according to [48] since we solve its relaxation optimally. be^{-b} is maximized when $b = 1$, so we use $(1, \frac{1}{e})$ contention resolution scheme to achieve the best ratio. This ratio becomes $\frac{1}{e} + (1 - \frac{1}{e})\overline{Q}$ for our minimization problem as derived in (6.20)–(6.24):

$$-\hat{Q}^n(X^r|Y) + A \leq -\hat{Q}^n(X^*|Y)\frac{1}{e} + A \tag{6.20}$$

$$Q(X^r|Y) \leq -\left(-Q(X^*|Y) + A\right)\frac{1}{e} + A \tag{6.21}$$

$$Q(X^r|Y) \leq Q(X^*|Y)\frac{1}{e} + A(1 - \frac{1}{e}) \tag{6.22}$$

$$Q(X^r|Y) \leq Q(X^*|Y)\left(\frac{1}{e} + \left(1 - \frac{1}{e}\right)\frac{\sum_{(u,v)\in V^2} F_{u,v}^2 + \sum_{i\in I}\sum_{(p,d)\in V_q} w_{pd}^c}{Q(X^*|Y)}\right) \tag{6.23}$$

$$Q(X^r|Y) \leq Q(X^*|Y)\left(\frac{1}{e} + \left(1 - \frac{1}{e}\right)\underbrace{\frac{Q(\emptyset,\emptyset) + \epsilon}{k\min_{(p,d)}(w_{pd}^c) + \epsilon}}_{\overline{Q}}\right) \tag{6.24}$$

where $\overline{Q} = \frac{Q(\emptyset,\emptyset)+\epsilon}{k\min_{(p,d)}(w_{pd}^c)+\epsilon}$ for arbitrarily small constant $\epsilon > 0$ since we approximate the lower bound of the optimal solution by $Q(X^*|Y) \geq k\min_{(p,d)}(w_{pd}^c)$.

□

6.2.4 Step 2: SDP Relaxation of Binary Least Squares for Density Assignment

Given *BQC* assignments X, (6.14) disappears, and the resulting program is a binary quadratic program under the assignment constraints (6.15). However, the size of this program is linear

in terms of F_{max} which may be arbitrarily large. To efficiently estimate the class densities, we express the program more compactly by defining a variable for every $s \in S' = \{2^d \mid d \in 0, 1, \ldots, \lfloor \log(F_{max}) \rfloor\}$. This modification also removes (6.15) without losing any expressiveness since we can express any density up to F_{max} as a sum of subset of S'. The resulting problem is:

$$\min Q(Y|X) = \sum_{(u,v)\in V^2} \left(F_{u,v} - \left(\sum_{i\in I} m_{ui}\, m_{vi} \sum_{s\in S'} s y_{is} \right) \right)^2 + \text{Constant} \tag{6.25}$$

$$= \sum_{i\in I, s\in S'} \sum_{j\in I, t\in S'} st \Big(\sum_{(u,v)\in V^2} m_{ui} m_{vj} \Big) y_{is} y_{jt} - 2 \sum_{i\in I, s\in S'} s \Big(\sum_{(u,v)\in V^2} F_{u,v} m_{ui} m_{vi} \Big) y_{is}$$

where binary $y_{is} = 1$ if s is part of class i's density, m_{ui} is an indicator for whether u is assigned to a *BQC* in class i that is known from given X, and $\sum_{s\in S'} s y_{is}$ is the density of class i. Optimizing (6.25) is NP-hard via reduction from *PARTITION* [150]. To solve it efficiently, we turn our $\{0, 1\}$ quadratic program into homogenous $\{\pm 1\}$ quadratic program by replacing every y_{is} with $(1 + y'_{is})/2$ where $y'_{is} \in \{\pm 1\}$, and then by substituting $y'_{is} = r y''_{is}$ where $r \in \{\pm 1\}$. The resulting boolean program can be rewritten as:

$$\min_{Y} \quad \mathbf{y''TAy''} - 2\mathbf{b^T} r\mathbf{y''} + \|\mathbf{b}\|^2 \tag{6.26}$$

$$\text{s.t.} \quad y''^2_{is} = 1, \quad i \in 1, \ldots, k\,, s \in S' \tag{6.27}$$

$$r^2 = 1 \tag{6.28}$$

where $\mathbf{A}$ is the matrix of quadratic coefficients in (6.25) modified by the transformation above, $\mathbf{b}$ is the modified vector of linear coefficients in (6.25), and $\mathbf{y''}$ is a $k|S'|$ length vector. We relax this quadratically constrained quadratic program into the following semidefinite program (SDP):

$$\mathbf{Y}^* = \arg\min_{\mathbf{Y''}} Tr(\hat{\mathbf{A}}\mathbf{Y''}) \tag{6.29}$$

$$\text{s.t} \quad Y''^2_{t,t} = 1, \quad t \in 1, \ldots, k|S^{'}| + 1 \tag{6.30}$$

$$\mathbf{Y''} \succeq 0 \tag{6.31}$$

where $\mathbf{Y''} = [\mathbf{y^{T''}}, r]^T [\mathbf{y''}, r]$ is positive-semidefinite matrix, and $\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & -\mathbf{b} \\ -\mathbf{b^T} & \|\mathbf{b}\|^2 \end{bmatrix}$. After solving this SDP optimally, we run the following rounding procedure based on Gaussian sampling [98]: We generate a set of random vectors ξ_l, $l \in 1, \ldots, L = 100$ from multivariate Gaussian distribution $\mathcal{N}(0, \mathbf{Y}^*)$, quantize each of them into a binary vector $\hat{y}_l = \text{sign}(\xi_l) \in \{\pm 1\}^{k|S'|+1}$, and obtain a solution by $\hat{y} = \min_{l\in 1,\ldots,L} \hat{y}_l^T \mathbf{A} \hat{y}_l$. This procedure gives $\frac{2}{\pi} + (1 - \frac{2}{\pi})\overline{Q}$ approximation guarantee for Step 2 as proven in Lemma 6.2.4.

Lemma 6.2.4. *Step 2 can be approximated to a factor* $\frac{2}{\pi} + (1 - \frac{2}{\pi})\overline{Q}$.

Proof. Similar to the proof of Lemma 6.2.3, we turn the original minimization problem to nonnegative maximization problem. This nonnegative maximization problem can be approximated by $\frac{2}{\pi}$ since $-\hat{\mathbf{A}}$ in the objective is positive-semidefinite as the new objective always takes nonnegative values for any vector [98, 106]. This ratio is achieved by rounding the optimal SDP

solution via Gaussian sampling as described in Section 6.2.4. This approximation ratio for non-negative maximization becomes $\frac{2}{\pi} + (1 - \frac{2}{\pi})\overline{Q}$ for Step2 where $\overline{Q} = \frac{Q(\emptyset,\emptyset)+\epsilon}{k \min_{(p,d)}(w^c_{pd})+\epsilon}$ by following a similar derivation of Lemma 6.2.3.

□

6.2.5 The case of real-valued densities: *3CDEfrac*

We modify only Step 2 of *3CDEint* for nonnegative, real-valued class densities. Let y_i be the variable for class i's density, *3CDEfrac*'s second step optimally solves the following convex quadratic program:

$$\min_{Y} \sum_{i\in I} \sum_{j\in I} \Big(\sum_{(u,v)\in V^2} m_{ui} m_{vj} \Big) y_i y_j \; - 2 \sum_{i\in I} \Big(\sum_{(u,v)\in V^2} F_{uv} m_{ui} m_{vi} \Big) y_i \tag{6.32}$$

$$y_i \geq 0, \quad i \in I \tag{6.33}$$

6.3. Exact 3C Deconvolution Methods

For smaller problem instances, we develop optimal methods *3CDEint*-opt and *3CDEfrac*-opt based on convex Quadratic Integer Programming (QIP). *3CDEint*-opt can be expressed as in Program (6.34)–(6.39):

$$\min \sum_{(u,v)\in V^2} \left(F_{u,v} - \Big(\sum_{p\in M(u)\cap M(v)} \sum_{d=|u-v|}^{l_p-1} \sum_{i\in I} y_{pdi} \Big) \right)^2 - \sum_{i\in I} \sum_{(p,d)\in V_q} w^c_{pd} x_{pdi} \tag{6.34}$$

$$\text{s.t. } x_{pdi} + x_{rti} \leq 1, \quad \forall\, ((p,d),(r,t)) \in E_q, \forall i \in I \tag{6.35}$$

$$y_{pdi} \leq F_{max}\, x_{pdi}, \quad \forall (p,d) \in V_q, \forall i \in I \tag{6.36}$$

$$|y_{pdi} - y_{rti}| \leq F_{max}(2 - x_{pdi} - x_{rti}), \quad \forall\, ((p,d),(r,t)) \notin E_q, \forall i \in I \tag{6.37}$$

$$x_{pdi} \in \{0,1\}, \quad \forall (p,d) \in V_q, \forall i \in I \tag{6.38}$$

$$y_{pdi} \in \{0,1,\ldots,F_{max}\}, \quad \forall (p,d) \in V_q, \forall i \in I \tag{6.39}$$

where binary $x_{pdi} = 1$ if d-*BQC* of domain p is assigned to class i, and integer y_{pdi} is its density. Objective (6.34) is convex as shown previously, and overlapping *BQC*s cannot coexist in the same class according to (6.35). Constraints (6.36) ensure that density of d-*BQC* of domain p in class i is 0 if not used in i, and if assigned, its density is at most F_{max}. Lastly, (6.37) ensures that all *BQC*s of the same class have the same density. When the class densities are real-valued, we propose *3CDEfrac*-opt by relaxing the integer density constraints (6.39) in Program (6.34)–(6.39) which turns it into convex Mixed Integer Quadratic Program (MIQP).

6.4. Results

6.4.1 Implementation

We implemented our methods using CPLEX [145] to solve LP, ILP and convex quadratic programs, and SDPT3 [143] to solve SDP relaxations. We use the public implementations of *Armatus* [50] and *MCMC5C* [119] for comparison, and implemented the 3C normalization method by [156]. Code and data are available on the web[1]. The approximate methods are reasonably fast: *3CDEint* and *3CDEfrac* can deconvolve *CD*4^+ interaction matrices in less than 15 minutes on a laptop with 2.5Ghz processor and 8Gb Ram when $l_{\max} = 25$. They typically converge in fewer than 5 iterations. Our methods can also deconvolve larger 20-40 kbp resolution matrices under 30 minutes by restricting $l_{\max} = 50$ as topological domains are a few megabases in length.

6.4.2 Evaluating Performance

We evaluate deconvolution methods in the few cases where small, synchronized populations were assayed with 3C methods. Nagano et al. [104] performed Hi-C on 10 single mouse cells, Naumova et al. [105] performed Hi-C on several populations HeLa cells, each synchronized to a specific phase of the cell cycle, and Le et al. [87] performed Hi-C on populations of *Caulobacter* cells, also synchronized to various phases of the cell cycle. In each of these experiments, we have more-than-usual confidence that the assayed cells represent a single, unmixed population of structures. To simulate a more typical population of cells with mixture, we sum together the individual matrices from each of these experiments to obtain a synthetic ensemble matrix $\mathbf{F}$ that we then attempt to deconvolve into its constituent components (the matrices from the single cell or synchronized experiments). In each experiment, we form the interaction matrices by binning the raw interaction data at a given resolution where the value of entry (i, j) is the total number of interactions between the restriction sites in bins i and j. We analyze *CD*4^+'s each chromosome independently, but analyze the prokaryotic single chromosome of *Caulobacter* and only the 21'st chromosome of *HeLa* cells. When necessary, we remove the experimental 3C biases by normalizing the interaction data via [156].

We measure the agreement between our estimated subpopulation contact matrices and the true contact matrices (single cell or synchronized cell cycle) using two metrics: the normalized mean absolute error (MAE) and the normalized Variation of Information (NVI) [99]. Let $T_p = \{T_p^1, \ldots, T_p^k\}$ and $E_p = \{E_p^1, \ldots, E_p^k\}$ be the set of true and estimated domain partitions respectively, and $\mathbf{T}$ and $\mathbf{E}$ be the set of associated interaction matrices. To estimate either metric (MAE or VI), we perform a minimum-weight bipartite perfect matching between $\mathbf{T}$ and $\mathbf{E}$ where the edges are weighted by the value of the metric (VI or MAE) and the value of the agreement between $\mathbf{T}$ and $\mathbf{E}$ is the average value of the minimum perfect matching.

Variation of Information (VI) measures the similarities between two partitions, and it is normalized by dividing by $\log(n)$. Let $c_1 = [s_1, e_1]$ be an interval between s_1 and e_1, and $X = \{c_1, \ldots, c_a\}$ be a partition such that $\sum_{c \in X} (e_c - s_c + 1) = n$ where each c_1 represent either a domain or an inter-domain region, and none of c_i, c_j pairs overlap $[s_i, e_i] \cap [s_j, e_j] = \emptyset$. Given

[1] `http://www.cs.cmu.edu/~ckingsf/research/3cde`

partitions X and Y, $VI(X,Y) = H(X) + H(Y) - 2I(X,Y)$. $H(X) = -\sum_{c\in X} \log(p_c)\, p_c$ is the entropy of the partition X where $p_c = \frac{e_c - s_c + 1}{n}$ is the probability of seeing the interval $[s_c, e_c]$, and $I(X,Y) = \sum_{c_x \in X} \sum_{c_y \in Y} p_{x,y} \log(\frac{p_{x,y}}{p_x p_y})$ is the mutual information between X and Y where $p_{x,y} = \frac{[s_x,e_x] \cap [s_y,e_y]}{n}$. Given matrices $\mathbf{T^i}$ and $\mathbf{E^j}$, normalized mean absolute error (MAE) is defined as $MAE(\mathbf{T^i}, \mathbf{E^j}) = \frac{\sum_{u\in V} \sum_{v\in V} |T^i_{u,v} - E^j_{u,v}|}{n^2}$. True interaction matrices $\mathbf{T}$ are known, whereas true domain decompositions T_p are unknown so we define *consensus Armatus* domains of $\mathbf{T}$ as the truth. Lower score means better performance in both scores. In the case of VI, this metric measures agreement between clusterings (here partitions of fragments into domains and non-domains). Since the true domain partitions are unknown, we use the consensus *Armatus* domains computed on each known subpopulation as the truth. In both measures, lower score means better performance.

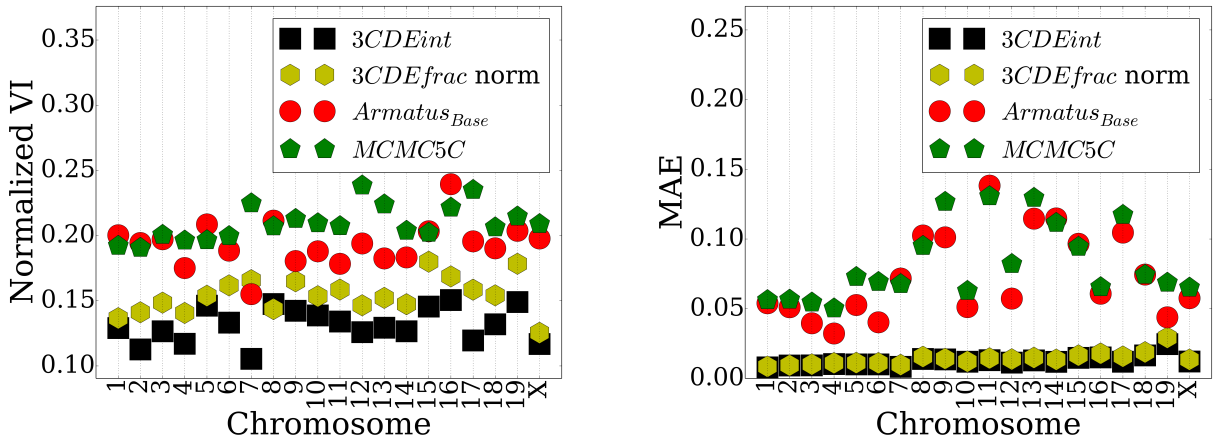
We compare our methods with greedy baseline *Armatus$_{Base}$* and *MCMC5C* [119]. In baseline *Armatus$_{Base}$*, we add the domains from the top-k *Armatus* decompositions into a set. For each class, we shuffle the set, and iterate through half of the set by assigning a domain from this set unless it intersects with the currently-assigned domains. We repeat this procedure 10000 times to estimate the distribution of the scores. Using domains from *Armatus* equips *Armatus$_{Base}$* with domains that appear in the convoluted data set, and it is therefore a more conservative comparison to our methods. We present the mean *Armatus$_{Base}$* score, and estimate P-values of our results from this distribution to test for the significance. We also estimate the matrices of k embeddings via inverse frequency-distance mapping in *MCMC5C*. When estimating the marker distribution, we define a domain boundary as a region extended to left and right of the exact boundary by half of the resolution since this reflects the uncertainty in its position due to binning. Unless otherwise noted, we use an exponential kernel for *BQC* quality, and assume no prior domain knowledge.

6.4.3 Deconvolution of Single Mouse *CD*4^+ Interaction Matrices

We apply our method and the baseline methods to the *CD*4^+ interaction dataset at 250 kbp resolution by providing them with the sum of the matrices from the 10 experiments in which 3C contacts were estimated on single mouse *CD*4^+ cells. We compare the estimated subpopulation matrices using this summed matrix as input to the original single cell matrices. Performance is shown in Figures 6.3a–6.3b.

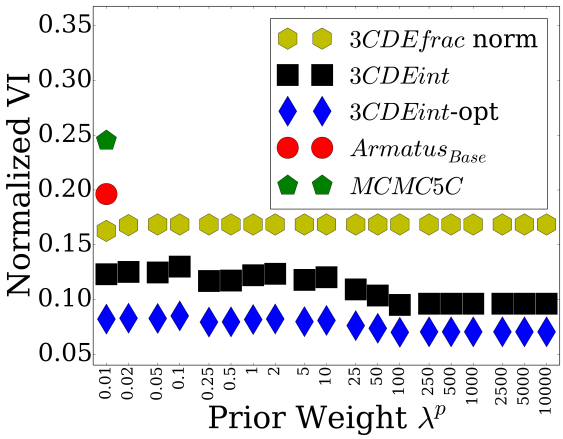
3CDEint and *3CDEfrac* nearly always perform the best in identifying contact matrices that match the single cell matrices. Even though *Armatus$_{Base}$* greedily assigns domains to the classes, mean *Armatus$_{Base}$* performs better than *MCMC5C* in Figure 6.3a for most of the chromosomes. *3CDEfrac* over normalized data [156] may perform worse than *Armatus$_{Base}$* because *CD*4^+ data is an ensemble over only 10 cells rather than millions of cells as in traditional 3C experiments. We observe a similar performance trend in terms of the metric MAE as in Figure 6.3b. Normalization does not decrease the performance as it did for normalized VI in Figure 6.3a. *3CDEint* performs significantly better than *Armatus$_{Base}$* on all chromosomes ($p < 0.05$) in terms of both metrics since variance of the distribution of *Armatus$_{Base}$* scores is low even though the mean scores are close to ours. In general, lower matrix error scores show the quality of the deconvolution in estimating the mixing matrices.

We examine the performance of chromosome 17 as the domain prior weight λ is increased



(a) Chromosome-wise NVI

(b) Chromosome-wise MAE



(c) Chro. 17 NVI

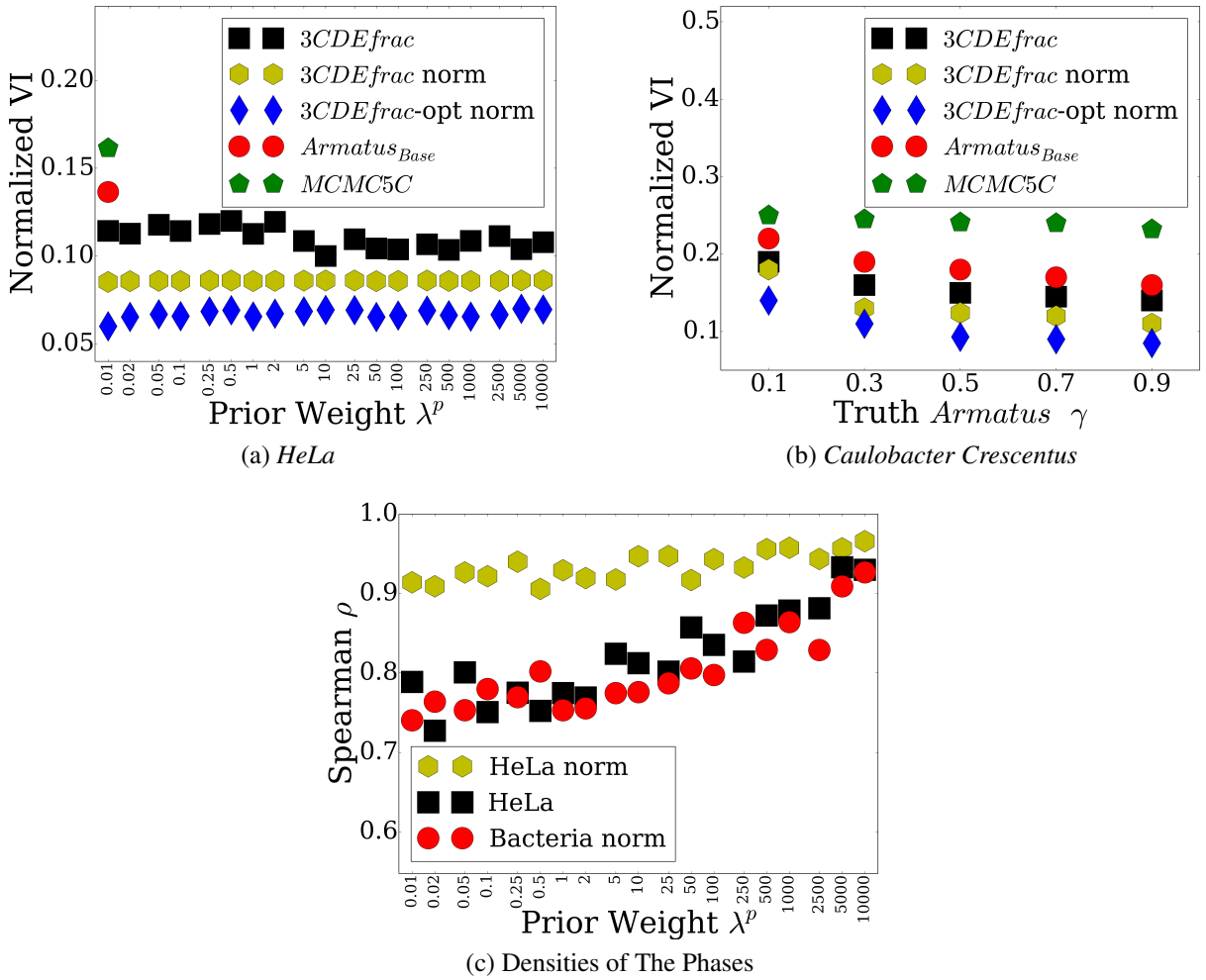
Figure 6.3: Chromosome-wise deconvolution performance of $CD4^+$ dataset in terms of (a) Normalized VI, (b) Mean Absolute Error (MAE). (c) Performance on the 17th chromosome for various prior weights λ^p.

(Figure 6.3c). The prior weight seems to have little effect on the overall performance, though *3CDEfrac* over normalized data is more robust to different prior weights. Chromosome 17 is small enough that we can use *3CDEint*-opt to find the true optimum of our objective (blue diamonds in Figure 6.3c). This shows that our heuristics are achieving close to the optimum value.

6.4.4 Temporal Deconvolution of Interphase Populations in *HeLa* and *Caulobacter* Cells

We deconvolve the sum of measured matrices of the 21st chromosome of *HeLa* cells at 250 kbp resolution using data from Naumova et al. [105]. Here, each subpopulation represents cells at a particular phase of the cell cycle, and so we are deconvolving along the temporal dimension. Figure 6.4a shows the performance for several choices of prior. Again, we match the true matrices

better than either a greedy approach or sampling approach (*MCMC5C*). All the methods perform better in *HeLa* cells than *CD*4^+ cells as shown in Figure 6.3c. Unlike in *CD*4^+, normalization improves the deconvolution performance as well as making the performance of both approximate *3CDEfrac* and exact *3CDEfrac*-opt less dependent on the prior weight. This performance stability shows that we may obtain true domain decompositions without strong reliance on prior data. *3CDEfrac* and *3CDEfrac*-opt also outperform the competing methods in terms of average error per matrix entry: *3CDEfrac* without a domain prior can achieve MAE of 0.004, whereas *MCMC5C* achieves almost 8-fold more MAE, 0.03.



(a) *HeLa*

(b) *Caulobacter Crescentus*

(c) Densities of The Phases

Figure 6.4: (a) Deconvolution performance on *HeLa* dataset by increasing prior weight λ^p in terms of NVI. (b) Performance on prokaryotic bacteria dataset vs. *Armatus* γ in terms of NVI. (c) Performance of *3CDEfrac* in estimating the densities of the cell cycle phases on eukaryotic *HeLa* and prokaryotic *Caulobacter* datasets in terms of Spearman's correlation ρ by increasing λ^p.

We performed a similar experiment for the bacterium *Caulobacter* where Le et al. (2013) provide cell-cycle-phase-specific Hi-C matrices. Figure 6.4b reports these results using the NVI

metric as the resolution of the ground truth domains was varied. While ground truth matrices are known in these experiments, the true domain decomposition is estimated computationally via a topological domain finder *Armatus*. This program has a parameter γ that controls the domain sizes, with larger γ corresponding to smaller domains. As γ increases, all methods perform better, however, the ranking of the methods in terms of performance is same regardless of γ. We observe similar performance trend on *HeLa* dataset as well. This shows both that we can deconvolve bacterial Hi-C experiments and that the performance is robust to the scale at which we define the true domains.

Our methods also estimate the densities of the mixing cell cycle phases quite accurately on *HeLa* and *Caulobacter* if densities of the 4 cell cycle phases (early G1, mid G1, S, M) are assumed to be proportional to their durations. Figure 6.4c plots the Spearman's ρ correlation between estimated and true densities at 250 kbp for both datasets. We often achieve correlations over 0.75. Existing methods do not provide any estimate of the densities of the subpopulations.

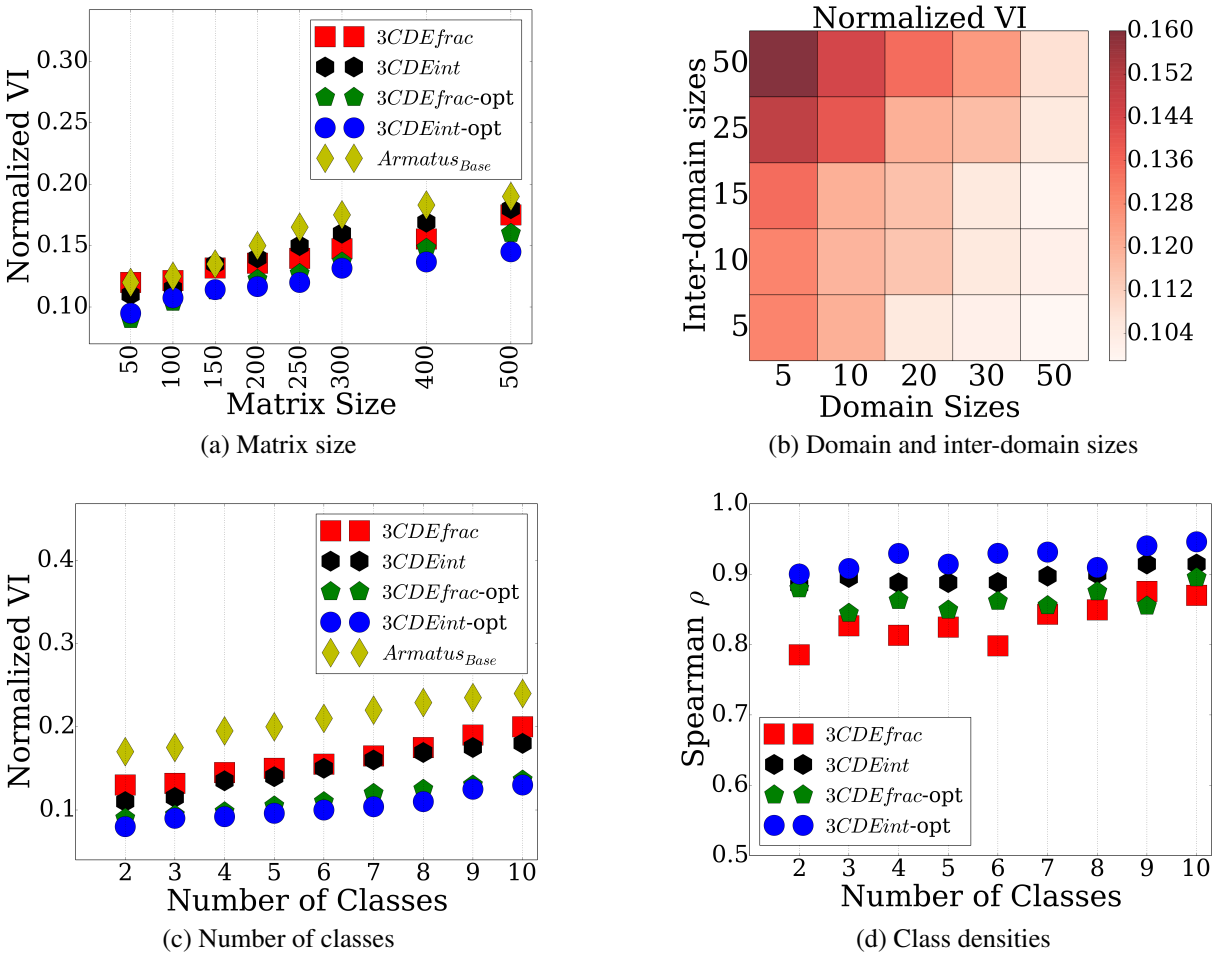
6.4.5 Results on Synthetic Interaction Data

To understand the practical hardness of the deconvolution problem under different types of class densities and wide range of domain sizes, we also tested our methods on synthetic data. There is no known domain generation procedure that mimics the true domain structure, so we generated the synthetic data as follows: For given number of classes and matrix sizes, in each class, we repeatedly flip an unbiased coin starting from the first bin to generate either domains of size sampled from gaussian distribution $\mathcal{N}(\mu = 40, \sigma^2 = 10)$ or $\mathcal{N}(\mu = 10, \sigma^2 = 4)$, or inter-domain regions of size sampled from $\mathcal{N}(\mu = 5, \sigma^2 = 1)$ until we reach the last bin. Similarly, we sample the class densities from $\mathcal{N}(\mu = 5, \sigma^2 = 2)$ by rounding them when the class densities are supposed to be integers. Lastly, we obtain the ensemble matrix by summing up the interaction matrices multiplied by their densities.

According to Figure 6.5a, increasing the matrix size by sampling the domain sizes from $\mathcal{N}(\mu = 10, \sigma^2 = 4)$ and inter-domain sizes from $\mathcal{N}(\mu = 5, \sigma^2 = 1)$ decreases the performance similar to effect of the increasing resolution on real datasets as in Figure (6.6a)-(6.6b). Increasing the matrix size also increases the performance difference between our methods and *Armatus*$_{Base}$. The ratio of the domain sizes to inter-domain sizes is the major determinant of the performance as in Heatmap 6.5b for *3CDEint*: Increasing the inter-domain sizes without increasing the domain sizes leads to poorer performance due to increasing number of possible optimal solutions. We also observe similar results for other methods. Figure 6.5c shows how deconvolution performance decreases by increasing number of classes for both approximate and exact methods under both integer and nonnegative densities. Lastly, our methods can also estimate the mixing class densities quite accurately in terms of Spearman's correlation ρ as in Figure 6.5d without being affected by the number of classes. Unlike the mixing matrices estimation, exact and approximate methods perform similarly in estimating the densities.

6.4.6 Effect of Resolution and Robustness Prior

The deconvolution methods developed here work well at various 3C resolutions. When we binned the input 3C matrices at increasing intervals, increasing the resolution leads to larger,



(a) Matrix size

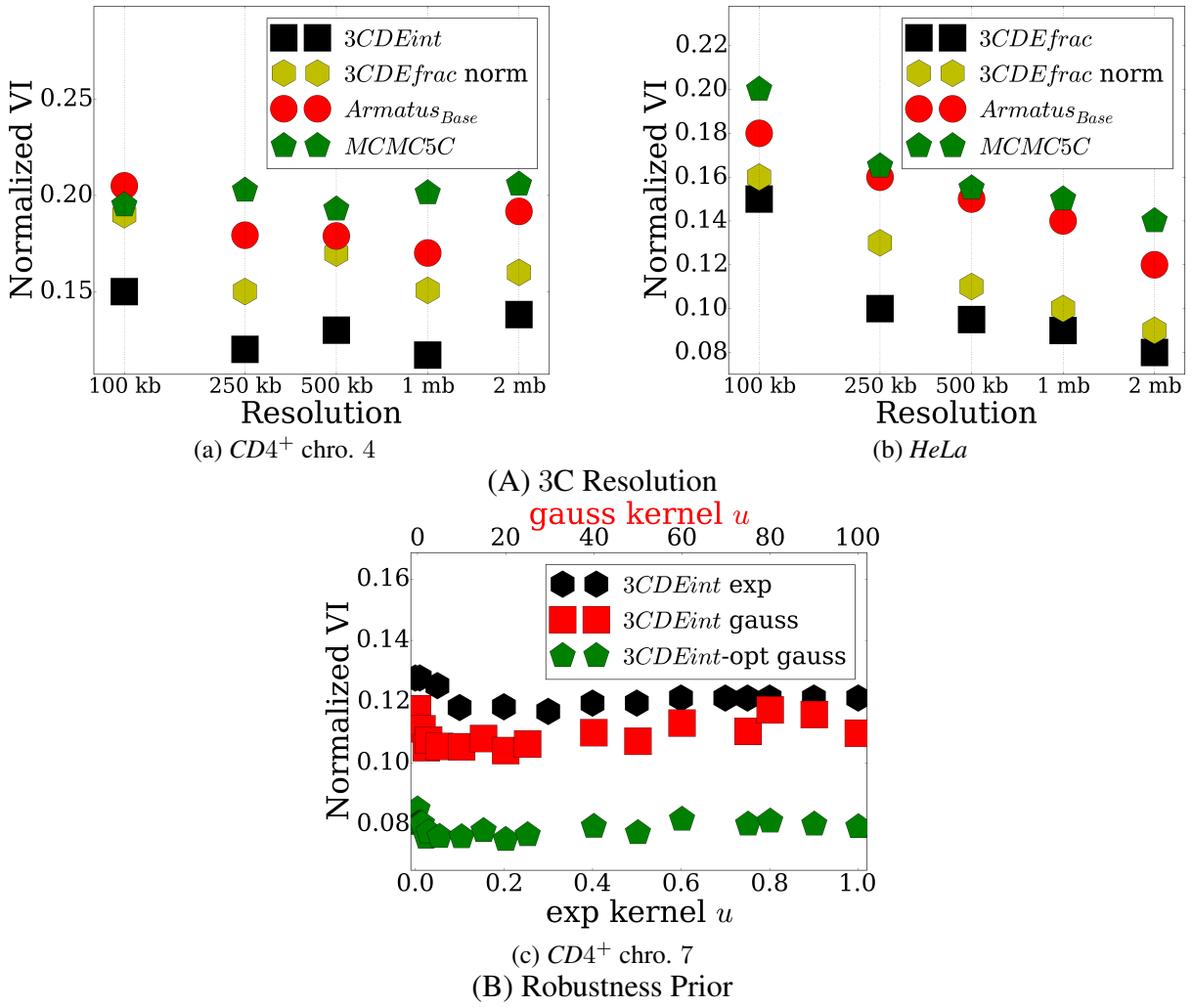
(b) Domain and inter-domain sizes

(c) Number of classes

(d) Class densities

Figure 6.5: Performance of our methods on synthetic dataset vs. a) interaction matrix sizes, b) domain and inter-domain sizes, c) number of classes in terms of Normalized VI; and d) class densities estimation performance in terms of Spearman's correlation ρ.

more detailed interaction matrices, which usually decreases the performance somewhat (Figure (6.6a)–(6.6b)). The performance decreases monotonically on *HeLa* dataset by increasing resolution, but the score trend is non-monotonic in $CD4^+$ cells due to its smaller population size with more influential outliers. However, the *3CDEfrac* and *3CDEint* methods still outperform the other methods. This is likely due in part to the definition of *BQC*s, which can properly model long-range, out-of-domain interactions in the higher resolution matrices. The choice of the kernel for the robustness prior also seems to have relatively little effect on performance as shown in Figure (6.6c) or the 7th $CD4^+$ chromosome. We obtain similar results for 21st *HeLa* chromosome.



(a) $CD4^+$ chro. 4

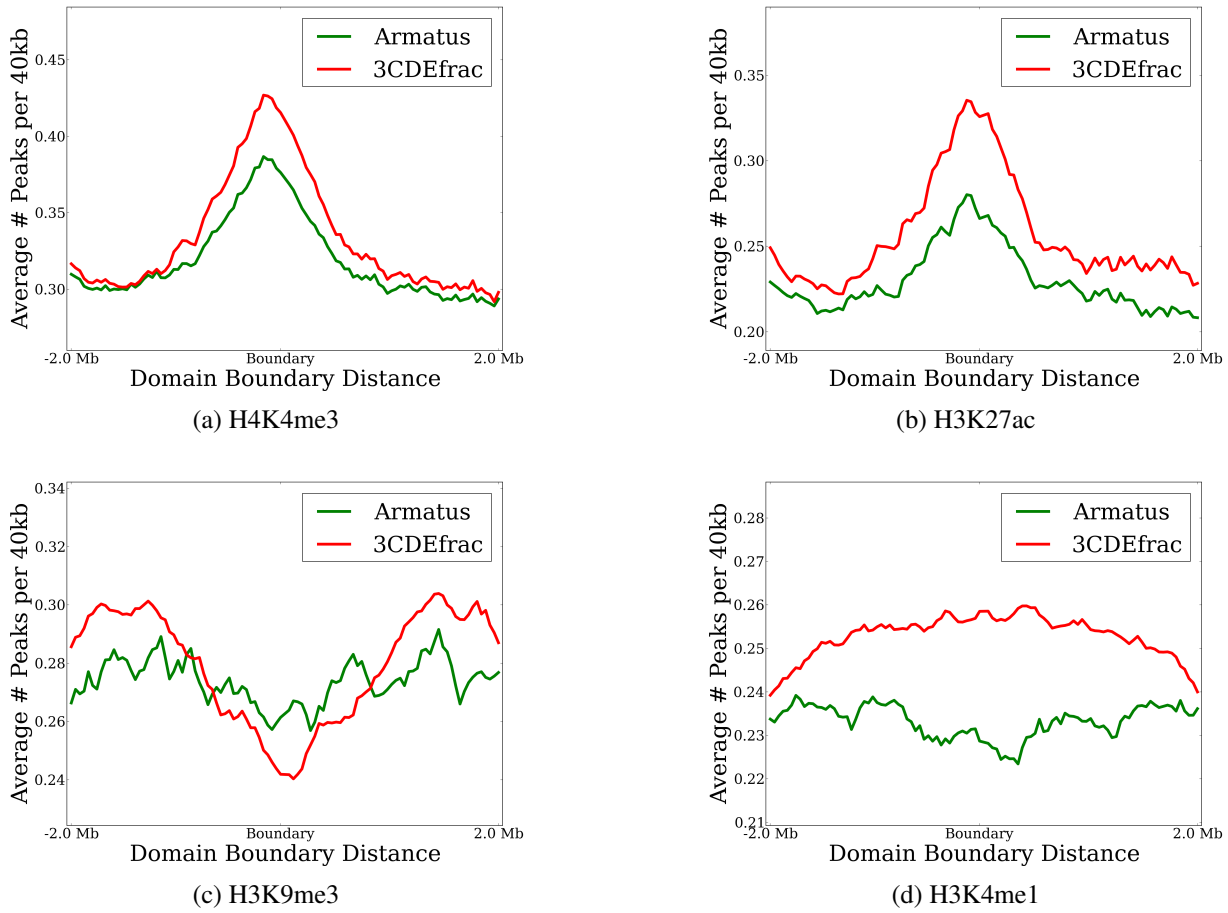
(b) *HeLa*

(c) $CD4^+$ chro. 7

Figure 6.6: Effect of 3C resolution on the performance in (a) 4th $CD4^+$ chromosome, (b) *HeLa* cells, and the effect of weighting kernel of the robustness prior in (c) $CD4^+$ chromosome 7.

6.4.7 Distribution of Epigenetic Markers Relative To Deconvolved Domains

Epigenetic markers are distributed differently in the genome depending on its conformation, and domain organization of the genome is correlated to a certain extent with their distribution. For instance, H3K4me3 and CTCF binding sites are enriched in the domain boundaries due to their insulator roles. We calculate the distribution of several such markers near the domain boundaries as identified within the subpopulation matrices (Figure (6.7)–(6.8)). Each subfigure in Figure (6.7)–(6.8) plots the average number of markers in 40 kb bins for $+/-$ 2 Mb from all the estimated domain boundaries that occur within some estimated subpopulation matrix. For *Armatus* domain, we estimate the average number of markers over top-k decompositions for multiple γ between 0.1 and 0.9 ($k = 4$ for *HeLa*, and $k = 10$ for $CD4^+$). We obtain histone markers H3K4me3, H3K4me1, H3K9me3, H3K27ac, H3K27me3 from ChIP-Seq experiments [30, 136]



(a) H4K4me3

(b) H3K27ac

(c) H3K9me3

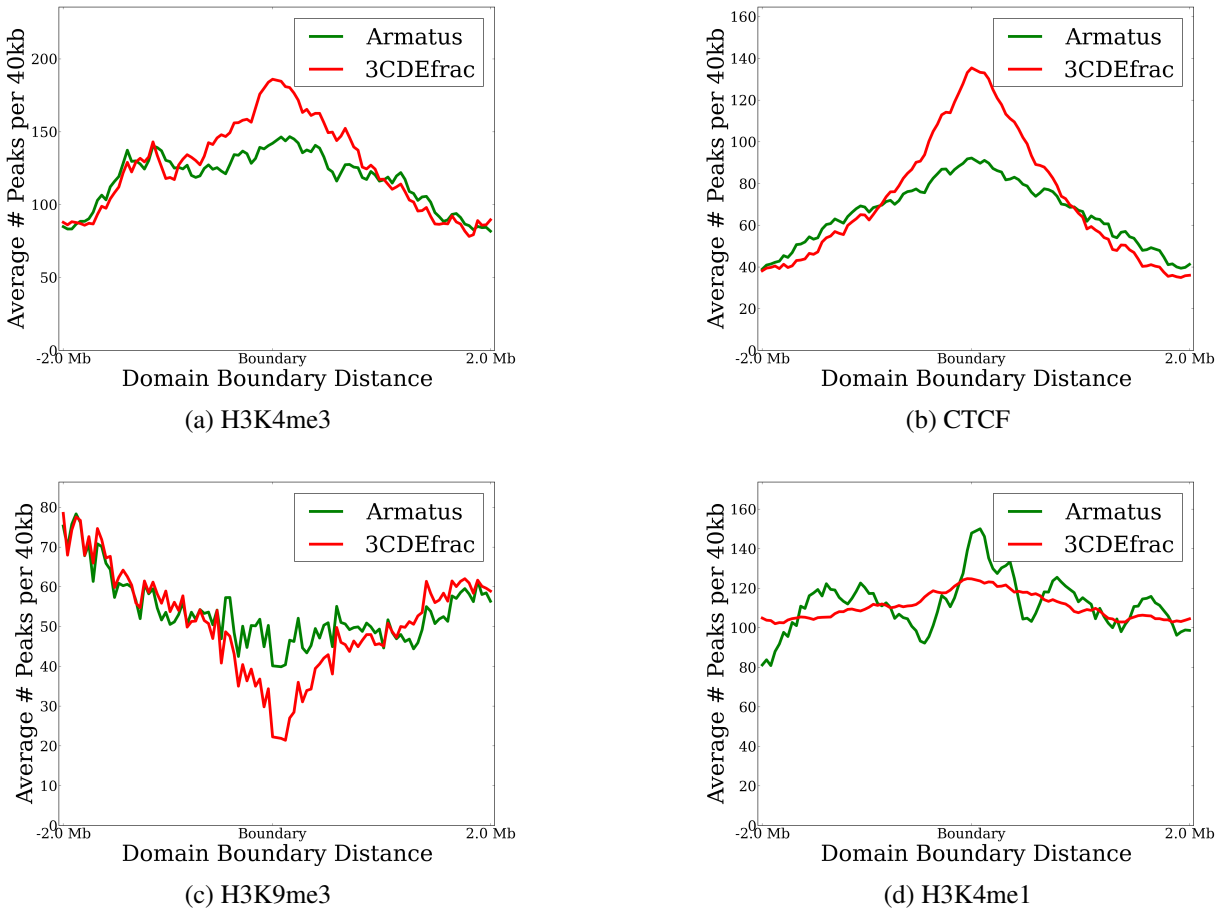
(d) H3K4me1

Figure 6.7: Distribution of several markers around the domain boundaries in $CD4^+$ cells. Red and green lines represent *3CDEfrac* and *Armatus* respectively in all plots.

for $CD4^+$ cells, from [11] for *HeLa* cells, and add CTCF sites from CTCFBSDB [160]. Mouse markers are not particularly for $CD4^+$ cells but instead they are detected over similar Th1 cells.

Overall, the relationship between histone markers and our domain boundaries are consistent with the experimentally-characterized different roles of the epigenetic markers [11]. Barrier-like histones H3K4me3, H3K27ac, and CTCF are more enriched in the deconvolved domain boundaries than *Armatus* boundaries in both species, whereas non-promoter-associated repressor H3K9me3 is more depleted in the deconvolved domain boundaries. This greater enrichment and depletion of the histones near the deconvolved domain boundaries, in accordance with the experimental results, show the improvement in extracting biologically-plausible domains from the ensemble data achieved by deconvolution.

To better interpret these scores, we estimate the significance of these coverage scores with respect to the random positioning of the same domains in terms of both enrichment and depletion by shuffling the domains 10000 times and keeping the markers fixed. We estimate the resulting p value by combining the multiple p values from different $CD4^+$ chromosomes by Fisher's method [100]. In Table 6.1, the bold entries represent significantly enriched (*) and



(a) H3K4me3

(b) CTCF

(c) H3K9me3

(d) H3K4me1

Figure 6.8: Distribution of several markers around the domain boundaries in *HeLa* cells.

depleted (**) markers ($p < 0.05$). Consistent with the previous results, H3K4me3, H3K27ac and CTCF with insulator roles are significantly enriched in the domain boundaries, whereas non-promoter-associated trimethylations H3K9me3 and H3K27me3 are depleted in the boundaries. Enrichments in *CD*4^+ cells do not mainly depend on whether we use the prior domain data, but the prior *Armatus* domains make the enrichment differences more pronounced in *HeLa* cells since *CD*4^+ results are average over all chromosomes representing the whole genome whereas *HeLa* dataset is composed of a single chromosome. Overall, we may use our methods as alternative domain finders returning multiple domain decompositions in the ensemble as suggested by significance of the results. The ratios in Table 6.1 and the associated p values show the compatibility of the estimated marker distributions relative to the domains with the known distributions. In Table 6.1, both the average coverage by the boundaries and average fraction of the markers present in the domains do not change substantially by using *Armatus* domains as a prior in *CD*4^+ cells showing the quality of our deconvolved domains in comparison to *Armatus* domains.

We also examined the distribution of the number of markers in the domain boundaries to analyze their relative strength in domain formation. We observe that frequency of boundary-enriched H3K4me3 in *3CDEfrac* domains follows almost a scale-free distribution $P(k) \sim k^{-\gamma}$

	Marker coverage by the domain boundaries				Marker coverage by the domains			
	HeLa		*CD*4^+		*HeLa*		*CD*4^+	
	No prior	With a prior	No prior	With a prior	No prior	With prior	No prior	With a prior
H3K4me3	0.23	$\mathbf{0.32}^*$	$\mathbf{0.58}^*$	$\mathbf{0.62}^*$	0.49	0.62	$\mathbf{0.75}^{**}$	0.78
H3K27me3	$\mathbf{0.22}^{**}$	$\mathbf{0.31}^{**}$	0.56	$\mathbf{0.57}^{**}$	0.51	0.65	0.75	$\mathbf{0.76}^*$
H3K27ac	0.27	$\mathbf{0.40}^*$	$\mathbf{0.62}^*$	0.62	0.55	$\mathbf{0.72}^*$	0.80	$\mathbf{0.81}^{**}$
H3K9me3	$\mathbf{0.15}^{**}$	$\mathbf{0.22}^{**}$	$\mathbf{0.51}^{**}$	$\mathbf{0.53}^{**}$	$\mathbf{0.34}^{**}$	0.45	$\mathbf{0.81}^*$	$\mathbf{0.83}^*$
CTCF	0.21	0.30	0.61	$\mathbf{0.62}^*$	$\mathbf{0.47}^{**}$	0.59	$\mathbf{0.77}^*$	0.77
PolII	0.25	$\mathbf{0.36}^*$	$\mathbf{0.62}^*$	$\mathbf{0.63}^*$	0.53	$\mathbf{0.68}^*$	$\mathbf{0.78}^{**}$	0.80

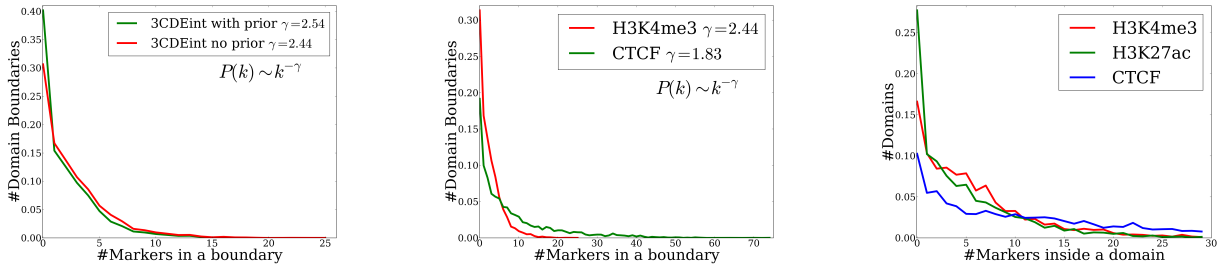
Table 6.1: The average fraction of the several markers in the domain boundaries and inside the domains extracted by *3CDEfrac* with and without *Armatus* domain prior in *HeLa* and *CD*4^+ cells. The bold entries represent significantly enriched (*) and depleted (**) markers ($p < 0.05$).

in CD4$^+$ cells as in Figure 6.9a by verifying its scale-freeness by log-log plot, without rejecting KS-test, and rejecting the hypotheses that it follows other possible exponential and log-normal distributions. Providing prior domain data slightly increases γ without affecting the shape of the frequency distribution. Similar frequency distributions of H3K4me3 in the boundaries by *3CDEfrac* with and without the prior shows the capabilities of our deconvolution methods as domain finders. Frequencies of several other markers, such as CTCF, are also close to power-law distributions as in Figure 6.9b. Higher γ of H3K4me3 reflects the fact that fewer number of highly insulating markers around a boundary is sufficient for a domain formation, whereas more of less barrier-like CTCF markers are needed in a domain boundary to form a separate domain. Different distributions of the markers reflect their different roles in the domain formation in agreement with our previous Figures (6.7)–(6.8). Scale-free distributions of the marker frequencies suggest the importance of the preferential attachment type mechanisms in topological domain formation which are greatly used to explain the scale-free degree distributions in real-world networks.

On the other hand, frequency distributions of the markers inside the domains do not follow a power law as in Figure 6.9c. H3K4me3 exists in abundance inside the domains similar to its abundant existence in the domain boundaries, whereas another insulator H3K27ac exists in fewer numbers inside the domains, and CTCF is seen inside almost every domain. Fewer number of H3K27ac markers inside the domains can be explained by its strong domain boundary termination feature which may have led to smaller domains if it has existed excessively inside the domains. Overall, our results show that many domains can be formed by smaller number of markers in their boundaries. We can consider domain formation as a complex interplay between the markers depending on their relative strengths. Figure 6.9 presents the results for CD4$^+$ cells, but frequency distributions in HeLa cells are similar.

6.5. Conclusion

We formulate the novel 3C deconvolution problem to estimate classes of contact matrices and their densities in the ensemble chromatin interaction data. We prove its hardness and design optimal and near-optimal methods that are practical on real data. Experimental results on mouse, *HeLa*, and bacteria datasets demonstrate that our methods outperform related methods in unmixing convoluted interaction matrices of prokaryotes and eukaryotes as well as in estimating the



(a) H3K4me3 (b) In the boundaries (c) Inside the domains

Figure 6.9: Distribution of the marker frequency in the domain boundaries (a,b) and inside domains (c) in CD4$^+$cells.

mixing densities without any biological prior. Our methods solve the previously unsolved problem of unmixing 3C experiments efficiently, and they return biologically meaningful domains supporting their alternative use as domain finders.

Chapter 7

Conclusions and Future Work

In this thesis, we propose solutions to four main problems related to biological and social networks and diffusion dynamics over them. We formulate all problems mathematically and apply them on different real-world datasets. In all problems, our approaches outperform the existing methods experimentally, and we discuss theoretical hardness of the problems under different settings.

In the first problem, we propose an optimization framework to predict protein annotations by using protein network data and the relationships between the protein functions. Our proposed method outperforms the existing methods on multiple species. In the second problem, we propose a network inference method with provable performance guarantees to estimate the unknown network from diffusion data at both micro and macro scales. In this case, our proposed method outperforms the competing approaches as well as returning novel diffusion estimates inside the United States. In the third problem, we propose scalable and fast methods to reconstruct diffusion histories from a limited number of diffusion snapshots under different diffusion models. Our methods can reconstruct the diffusion histories of several topics in social networks as well as identifying the initial spreaders of a contaminant in a water distribution network better than the previous approaches. Lastly, we propose methods to unmix the ensemble Hi-C data which we use to estimate the latent mixing matrices that represent cell subpopulations in the ensemble data. Unmixed matrices provide us insights about the relationship between histone distribution and Hi-C data. In almost all cases, our proposed method outperforms all the existing methods that cannot handle partial data. More detailed conclusion of each problem can be found on the respective chapters.

There are multiple directions for possible future work for each problem. Our solution to the protein annotation prediction problem on Chapter 3 can be further improved. One option is developing directed graphical models that can also take into account the directionality of the protein interaction network which data has recently started to become available [21, 151]. Another option is improving the current graph-based framework to integrate the protein sequence information.

Similarly, we can extend our methods for graph inference at both micro and macro scales on Chapter 4 to different diffusion models. Additionally, we assume that diffusion data is available apriori which may not be always satisfied in realistic settings. In this case, one must also consider the cost of collecting diffusion data, and active sampling approaches used for online learning in

different contexts [131, 133] can be adapted to our problem. Another open question regarding the inference problem is its theoretical hardness with respect to number of diffusion traces available. Even though theoretical hardness of the inference problem has been recently discussed for simpler diffusion models [1, 107], it is still open for many diffusion models.

We can also consider the diffusion history reconstruction problem on Chapter 5 under arbitrary diffusion models. Developing methods for more general diffusion models as well as discussing their theoretical hardnesses are still open problems. We can consider the problem for more general dynamics since existing dynamics may not be perfect for different diffusion types and different scales. Lastly, 3C deconvolution performance in Chapter 6 can be improved by integrating the microscopy data as a prior which is available for many species during cell cycles even though it is at low resolution.

Bibliography

[1] B. Abrahao, F. Chierichetti, R. Kleinberg, and A. Panconesi. Trace Complexity of Network Inference. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 491–499, New York, NY, USA, 2013. ACM.

[2] A. Agresti. *Categorical Data Analysis*. Wiley Series in Probability and Statistics. Wiley-Interscience, 2nd edition, 2002.

[3] R. M. Anderson and R. M. May. *Infectious Diseases of Humans Dynamics and Control*. Oxford University Press, 1992.

[4] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1-2):5–43, 2003.

[5] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, and G. M. R. . G. Sherlock. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25: 25–29, May 2000.

[6] F. Ay, E. M. Bunnik, N. Varoquaux, S. M. Bol, J. Prudhomme, J.-P. Vert, W. S. Noble, and K. G. Le Roch. Three-dimensional Modeling of the P. falciparum Genome During the Erythrocytic Cycle Reveals a Strong Connection Between Genome Architecture and Gene Expression. *Genome Research*, 24(6):974–988, 2014.

[7] N. T. J. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin London, 2nd edition, 1975.

[8] D. Balcan, V. Colizza, B. Gonalves, H. Hu, J. J. Ramasco, and A. Vespignani. Multiscale Mobility Networks and the Spatial Spreading of Infectious Diseases. *Proceedings of the National Academy of Sciences*, 106(51):21484–21489, 2009.

[9] Y. Bar-Yam. *Dynamics of Complex Systems*. Studies in Nonlinearity. Westview Press, 1997.

[10] A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286 (5439):509–512, 1999.

[11] A. Barski, S. Cuddapah, K. Cui, T.-Y. Roh, D. E. Schones, Z. Wang, G. Wei, I. Chepelev, and K. Zhao. High-Resolution Profiling of Histone Methylations in the Human Genome. *Cell*, 129(4):823 – 837, 2007.

[12] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya. Hierarchical Multi-label Prediction of Gene Function. *Bioinformatics*, pages 830–836, 2006.

[13] D. Baù and M. A. Marti-Renom. Structure Determination of Genomic Domains by Satisfaction of Spatial Restraints. *Chromosome Research*, 19(1):25–35, 2011.

[14] W. Bickmore and B. vanSteensel. Genome Architecture: Domain Organization of Interphase Chromosomes. *Cell*, 152(6):1270 – 1284, 2013.

[15] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[16] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[17] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-cut/Max-flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[18] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[19] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz. A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 649–658. IEEE, 2012.

[20] A. Budanitsky and G. Hirst. Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures. In *Workshop On WordNet and Other Lexical Resources, Second Meeting Of The North American Chapter Of The Association For Computational Linguistics*, 2001.

[21] M. Cao, C. M. Pietras, X. Feng, K. J. Doroschak, T. Schaffner, J. Park, H. Zhang, L. J. Cowen, and B. J. Hescott. New Directions for Diffusion-based Network Prediction of Protein Function: Incorporating Pathways with Confidence. *Bioinformatics*, 30(12):i219–i227, 2014.

[22] S. Carbon, A. Ireland, C. J. Mungall, S. Shu, B. Marshall, S. Lewis, et al. AmiGO: Online Access to Ontology and Annotation Data. *Bioinformatics*, 25(2):288–289, 2009.

[23] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. A Linear Programming Formulation and Approximation Algorithms for the Metric Labeling Problem. *SIAM Journal on Discrete Mathematics*, 18(3):608–625, 2004.

[24] L. L. Chen, N. Blumm, N. A. Christakis, A. L. Barabási, and T. S. Deisboeck. Cancer Metastasis Networks and the Prediction of Progression Patterns. *British Journal of Cancer*, 101(5):749–58, 2009.

[25] J. Cheng, M. Cline, J. Martin, D. Finkelstein, T. Awad, D. Kulp, and M. A. Siani-Rose. A Knowledge-based Clustering Algorithm Driven by Gene Ontology. *Journal of Biopharmaceutical Statistics*, 14(3):687–700, 2004.

[26] E. Cho, S. A. Myers, and J. Leskovec. Friendship and Mobility: User Movement in

Location-based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1082–1090, New York, NY, USA, 2011. ACM.

[27] J. Chuzhoy and J. S. Naor. The Hardness of Metric Labeling. In *Proceedings of 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 108–114, Washington, DC, 2004. IEEE Computer Society.

[28] P. Collas. The Current State of Chromatin Immunoprecipitation. *Molecular Biotechnology*, 45(1):87–100, 2010.

[29] J. Costello, M. Dalkilic, S. Beason, J. Gehlhausen, R. Patwardhan, S. Middha, B. Eads, and J. Andrews. Gene Networks in Drosophila Melanogaster: Integrating Experimental Data to Predict Gene Function. *Genome Biology*, 10(9):R97, 2009.

[30] A. M. Deaton, S. Webb, A. R. Kerr, R. S. Illingworth, J. Guy, R. Andrews, and A. Bird. Cell Type-specific DNA Methylation at Intragenic CpG islands in the Immune System. *Genome Research*, 21(7):1074–1086, 2011.

[31] A. Defazio and T. S. Caetano. A Convex Formulation for Learning Scale-Free Networks via Submodular Relaxation. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1259–1267. 2012.

[32] J. Dekker. Gene Regulation in the Third Dimension. *Science*, 319(5871):1793–1794, 2008.

[33] J. Dekker, M. A. Marti-Renom, and L. A. Mirny. Exploring the Three-dimensional Organization of Genomes: Interpreting Chromatin Interaction Data. *Nature Reviews Genetics*, 14(6):390–403, 2013.

[34] M. Deng, Z. Tu, F. Sun, and T. Chen. Mapping Gene Ontology to Proteins Based on Protein–protein Interaction Data. *Bioinformatics*, 20(6):895–902, 2004.

[35] J. R. Dixon, S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J. S. Liu, and B. Ren. Topological Domains in Mammalian Genomes Identified by Analysis of Chromatin Interactions. *Nature*, 485(7398):376–380, 2012.

[36] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press, 2003.

[37] D. Dotan-Cohen, S. Kasif, and A. A. Melkman. Seeing the Forest for the Trees: Using the Gene Ontology to Restructure Hierarchical Clustering. *Bioinformatics*, 25(14):1789–1795, 2009.

[38] N. Du, L. Song, A. J. Smola, and M. Yuan. Learning Networks of Heterogeneous Influence. In *NIPS*, pages 2789–2797, 2012.

[39] Z. Duan, M. Andronescu, K. Schutz, S. McIlwain, Y. J. Kim, C. Lee, J. Shendure, S. Fields, C. A. Blau, and W. S. Noble. A Three-dimensional Model of the Yeast Genome. *Nature*, 465(7296):363–367, 2010.

[40] G. Duggal, H. Wang, and C. Kingsford. Higher-order Chromatin Domains Link eQTLs With the Expression of Far-away Genes. *Nucleic Acids Research*, 42(1):87–96, 2014.

[41] M. S. Elmohamed, D. Kozen, and D. R. Sheldon. Collective Inference on Markov Models for Modeling Bird Migration. In *Advances in Neural Information Processing Systems 20*, pages 1321–1328. 2007.

[42] P. Erdös and A. Rényi. On the Evolution of Random Graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[43] J. Ernst and M. Kellis. ChromHMM: Automating Chromatin-state Discovery and Characterization. *Nature Methods*, 9(3):215–216, 2012.

[44] J. Fakcharoenphol, S. Rao, and K. Talwar. A Tight Bound on Approximating Arbitrary Metrics by Tree Metrics. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 448–455, New York, NY, USA, 2003. ACM.

[45] U. Feige. A Threshold of $\ln n$ for Approximating Set Cover. *Journal of the ACM*, 45(4): 634–652, 1998.

[46] U. Feige and M. Goemans. Approximating the Value of Two Power Proof Systems, with Applications to MAX 2SAT and MAX DICUT. In *Proceedings of the 3rd Israel Symposium on the Theory of Computing Systems (ISTCS'95)*, ISTCS '95, pages 182–189, 1995.

[47] U. Feige, V. S. Mirrokni, and J. Vondrak. Maximizing Non-Monotone Submodular Functions. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 461–471, Washington, DC, USA, 2007. IEEE Computer Society.

[48] M. Feldman, J. S. Naor, and R. Schwartz. A Unified Continuous Greedy Algorithm for Submodular Maximization. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 0:570–579, 2011.

[49] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Language, speech, and communication. MIT Press, 1998.

[50] D. Filippova, R. Patro, G. Duggal, and C. Kingsford. Multiscale Identification of Topological Domains in Chromatin. In *Algorithms in Bioinformatics*, volume 8126 of *Lecture Notes in Computer Science*, pages 300–312. Springer Berlin Heidelberg, 2013.

[51] G. Fudenberg, G. Getz, M. Meyerson, and L. A. Mirny. High Order Chromatin Architecture Shapes the Landscape of Chromosomal Alterations in Cancer. *Nature Biotechnology*, 29(12):1109–1113, 2011.

[52] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. *Molecular Biology of the Cell*, 11(12):4241–4257, 2000.

[53] A.-C. Gavin, M. Bosche, R. Krause, et al. Functional Organization of the Yeast Proteome by Systematic Analysis of Protein Complexes. *Nature*, 415(6868):141–147, 2002.

[54] GLPK. GNU Linear Programming Kit, 2010. http://www.gnu.org/software/glpk/.

[55] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring Networks of Diffusion and Influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1019–1028, New York, NY, USA, 2010. ACM.

[56] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the Temporal Dynamics of Diffusion Networks. In *Proceedings of the 28th International Conference on Machine Learning*, pages 561–568, 2011.

[57] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring Networks of Diffusion and Influence. *ACM Transactions on Knowledge Discovery from Data*, 5(4):1–37, 2012.

[58] M. Gomez Rodriguez, J. Leskovec, and B. Schölkopf. Structure and Dynamics of Information Pathways in Online Media. WSDM '13, pages 23–32, New York, NY, USA, 2013. ACM.

[59] D. Gorkin, D. Leung, and B. Ren. The 3D Genome in Transcriptional Regulation and Pluripotency. *Cell Stem Cell*, 14(6):762 – 775, 2014.

[60] A. Gupta, A. Roth, G. Schoenebeck, and K. Talwar. Constrained Non-monotone Submodular Maximization: Offline and Secretary Algorithms. In *Proceedings of the 6th International Conference on Internet and Network Economics*, WINE'10, pages 246–257, Berlin, Heidelberg, 2010. Springer-Verlag.

[61] H. Hethcote. The Mathematics of Infectious Diseases. *SIAM Review*, 42(4):599–653, 2000.

[62] H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Takagi. Assessment of Prediction Accuracy of Protein Function from Protein–protein Interaction Data. *Yeast*, 18(6):523–531, 2001.

[63] Y. Ho, A. Gruhler, A. Heilbut, et al. Systematic Identification of Protein Complexes in *Saccharomyces Cerevisiae* by Mass Spectrometry. *Nature*, 415(6868):180–183, 2002.

[64] D. S. Hochbaum. Instant Recognition of Polynominal Time Solvability, Half Integrality and 2-approximations. In *Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, APPROX '00, pages 2–14. Springer-Verlag, 2000.

[65] M. Hu, K. Deng, Z. Qin, J. Dixon, S. Selvaraj, J. Fang, B. Ren, and J. S. Liu. Bayesian Inference of Spatial Organizations of Chromosomes. *PLoS Computational Biology*, 9(1): e1002893, 2013.

[66] W.-K. Huh, J. V. Falvo, L. C. Gerke, A. S. Carroll, R. W. Howson, J. S. Weissman, and E. K. O'Shea. Global Analysis of Protein Localization in Budding Yeast. *Nature*, 425 (6959):686–691, 2003.

[67] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A Comprehensive Two-hybrid Analysis to Explore the Yeast Protein Interactome. *Proceedings of the National Academy of Sciences*, 98(8):4569–4574, 2001.

[68] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data. *Science*, 302(5644):449–453, 2003.

[69] L. J. Jensen, R. Gupta, H.-H. Strfeldt, and S. Brunak. Prediction of Human Protein Function According to Gene Ontology Categories. *Bioinformatics*, 19(5):635–642, 2003.

[70] F. Jin, Y. Li, J. R. Dixon, S. Selvaraj, Z. Ye, A. Y. Lee, C.-A. Yen, A. D. Schmitt, C. A.

Espinoza, and B. Ren. A High-resolution Map of the Three-dimensional Chromatin Interactome In Human Cells. *Nature*, 503(7475):290–294, 2013.

[71] R. Kalhor, H. Tjong, N. Jayathilaka, F. Alber, and L. Chen. Genome Architectures Revealed by Tethered Chromosome Conformation Capture and Population-based Modeling. *Nature Biotechnology*, 30(1):90–98, 2012.

[72] U. Karaoz, T. M. Murali, S. Letovsky, Y. Zheng, C. Ding, C. R. Cantor, and S. Kasif. Whole-genome Annotation by Using Evidence Integration in Functional-linkage Networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2888–2893, 2004.

[73] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the Spread of Influence Through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.

[74] T. K. Kerppola. Bimolecular Fluorescence Complementation (BiFC) Analysis as a Probe of Protein Interactions in Living Cells. *Annual Review of Biophysics*, 37(1):465–487, 2008.

[75] J. Kivinen and M. K. Warmuth. Exponentiated Gradient Versus Gradient Descent for Linear Predictors. *Information and Computation*, 132(1):1 – 63, 1997.

[76] J. Kleinberg and E. Tardos. Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields. *J. ACM*, 49(5): 616–639, 2002.

[77] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *J. ACM*, 46(5): 604–632, 1999.

[78] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:65–81, 2004.

[79] N. Komodakis and G. Tziritas. Approximate Labeling via Graph Cuts Based on Linear Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8): 1436–1453, Aug 2007.

[80] Y. A. I. Kourmpetis, A. D. J. van Dijk, M. C. A. M. Bink, R. C. H. J. van Ham, and C. J. F. ter Braak. Bayesian Markov Random Field Analysis for Protein Function Prediction Based on Network Data. *PLoS ONE*, 5(2):e9293, 2010.

[81] M. D. Kui, K. Zhang, S. Mehta, T. Chen, and F. Sun. Prediction of Protein Function Using Protein-Protein Interaction Data. *Journal of Computational Biology*, 10:947–960, 2002.

[82] M. P. Kumar and D. Koller. MAP Estimation of Semi-metric MRFs via Hierarchical Graph Cuts. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 313–320, Arlington, Virginia, United States, 2009. AUAI Press.

[83] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[84] L. Lane, G. Argoud-Puy, A. Britan, I. Cusin, P. D. Duek, O. Evalet, A. Gateau, P. Gaudet, A. Gleizes, A. Masselot, C. Zwahlen, and A. Bairoch. neXtProt: A Knowledge Platform for Human Proteins. *Nucleic Acids Research*, 40(D1):D76–D83, 2012.

[85] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila. Finding Effectors in Social Networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '10*, page 1059, New York, New York, USA, July 2010. ACM Press.

[86] D. Lazer, R. Kennedy, G. King, and A. Vespignani. The Parable of Google Flu: Traps in Big Data Analysis. *Science*, 343(6176):1203–1205, 2014.

[87] T. B. K. Le, M. V. Imakaev, L. A. Mirny, and M. T. Laub. High-Resolution Mapping of the Spatial Organization of a Bacterial Chromosome. *Science*, 342(6159):731–734, 2013.

[88] H. Lee, Z. Tu, M. Deng, F. Sun, and T. Chen. Diffusion Kernel-based Logistic Regression Models for Protein Function Prediction. *Omics: A Journal of Integrative Biology*, 10(1): 40–55, 2006.

[89] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko. Non-monotone Submodular Maximization Under Matroid and Knapsack Constraints. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 323–332, New York, NY, USA, 2009. ACM.

[90] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs Over Time: Densification Laws, Shrinking Diameters and Possible Explanations. KDD '05, pages 177–187, New York, NY, USA, 2005. ACM.

[91] J. Leskovec, L. A. Adamic, and B. A. Huberman. The Dynamics of Viral Marketing. *ACM Transactions on the Web (TWEB)*, 1(1), 2007.

[92] J. M. Levsky and R. H. Singer. Fluorescence in Situ Hybridization: Past, Present and Future. *Journal of Cell Science*, 116(14):2833–2838, 2003.

[93] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, London, UK, 1995.

[94] E. Lieberman-Aiden, N. van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. Lajoie, P. Sabo, M. Dorschner, R. Sandstrom, B. Bernstein, M. Bender, M. Groudine, A. Gnirke, J. Stamatoyannopoulos, L. Mirny, E. Lander, and J. Dekker. Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. *Science*, 326(5950):289–293, 2009.

[95] D. Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[96] D. Lin. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*, COLING '98, pages 768–774, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[97] R. Liu, V. Duvvuri, and J. Wu. Spread Pattern Formation of H5N1-avian Influenza and Its Implications for Control Strategies. *Mathematical Modelling of Natural Phenomena*,

3(07):161–179, 2008.

[98] Z.-Q. Luo, W.-K. Ma, A.-C. So, Y. Ye, and S. Zhang. Semidefinite Relaxation of Quadratic Optimization Problems. *Signal Processing Magazine, IEEE*, 27(3):20–34, 2010.

[99] M. Meilă. Comparing Clusterings—An Information Based Distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[100] F. Mosteller and R. A. Fisher. Questions and Answers. *The American Statistician*, 2(5): pp. 30–31, 1948.

[101] K. P. Murphy. Hidden Semi-markov Models (HSMMs). Technical report, 2002.

[102] S. Myers and J. Leskovec. On the Convexity of Latent Social Network Inference. In *Advances in Neural Information Processing Systems 23*, pages 1741–1749. 2010.

[103] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome Prediction of Protein Function via Graph-theoretic Analysis of Interaction Maps. *Bioinformatics*, 21 (1):302–310, 2005.

[104] T. Nagano, Y. Lubling, T. J. Stevens, S. Schoenfelder, E. Yaffe, W. Dean, E. D. Laue, A. Tanay, and P. Fraser. Single-cell Hi-C Reveals Cell-to-cell Variability in Chromosome Structure. *Nature*, 502(7469):59–64, 2013.

[105] N. Naumova, M. Imakaev, G. Fudenberg, Y. Zhan, B. R. Lajoie, L. A. Mirny, and J. Dekker. Organization of the Mitotic Chromosome. *Science*, 342(6161):948–953, 2013.

[106] Y. Nesterov. Semidefinite Relaxation and Nonconvex Quadratic Optimization. *Optimization Methods and Software*, 9(1-3):141–160, 1998.

[107] P. Netrapalli and S. Sanghavi. Learning the Graph of Epidemic Cascades. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, pages 211–222, New York, NY, USA, 2012. ACM.

[108] M. Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

[109] M. E. J. Newman. Spread of Epidemic Disease on Networks. *Physical Review E*, 66(1): 016128, July 2002.

[110] M. E. J. Newman. Modularity and Community Structure in Networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[111] A. Ostfeld, J. G. Uber, E. Salomons, J. W. Berry, W. E. Hart, C. A. Phillips, J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan, et al. The Battle of Water Sensor Networks (BWSN): A Design Challenge for Engineers and Algorithms. *Journal of Water Resources Planning and Management*, 134(6):556–568, 2008.

[112] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos. Threshold Conditions for Arbitrary Cascade Models on Arbitrary Networks. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 537–546, Washington, DC, USA, 2011. IEEE Computer Society.

[113] B. A. Prakash, J. Vreeken, and C. Faloutsos. Spotting Culprits in Epidemics: How Many

and Which Ones? In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, ICDM '12, pages 11–20, Washington, DC, USA, 2012. IEEE Computer Society.

[114] O. Puig, F. Caspary, G. Rigaut, B. Rutz, E. Bouveret, E. Bragado-Nilsson, M. Wilm, and B. Sraphin. The Tandem Affinity Purification (TAP) Method: A General Procedure of Protein Complex Purification. *Methods*, 24(3):218 – 229, 2001.

[115] J.-C. Rain, L. Selig, H. De Reuse, V. Battaglia, C. Reverdy, S. Simon, G. Lenzen, F. Petel, J. Wojcik, V. Schächter, et al. The Protein–protein Interaction Map of Helicobacter Pylori. *Nature*, 409(6817):211–215, 2001.

[116] P. Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[117] M. G. Rodriguez and B. Schölkopf. Submodular Inference of Diffusion Networks from Multiple Trees. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 489–496, 2012.

[118] L. A. Rossman. The EPANET Programmers Toolkit for Analysis of Water Distribution Systems. In *ASCE 29th Annual Water Resources Planning and Management Conference*, pages 39–48, 1999.

[119] M. Rousseau, J. Fraser, M. Ferraiuolo, J. Dostie, and M. Blanchette. Three-dimensional Modeling of Chromatin Structure from Interaction Frequency Data Using Markov Chain Monte Carlo Sampling. *BMC Bioinformatics*, 12(1):1–16, 2011.

[120] M. Rousseau, J. L. Crutchley, H. Miura, M. Suderman, M. Blanchette, and J. Dostie. Hox in Motion: Tracking HoxA Cluster Conformation During Differentiation. *Nucleic Acids Research*, 42(3):1524–1540, 2014.

[121] M. Salathè, M. Kazandjieva, J. W. Lee, P. Levis, M. W. Feldman, and J. H. Jones. A high-resolution Human Contact Network for Infectious Disease Transmission. *Proceedings of the National Academy of Sciences*, 107(51):22020–22025, 2010.

[122] A. Sanyal, B. R. Lajoie, G. Jain, and J. Dekker. The Long-range Interaction Landscape of Gene Promoters. *Nature*, 489(7414):109–113, Sep 2012.

[123] A. Schlicker, F. Domingues, J. Rahnenfuhrer, and T. Lengauer. A New Measure for Functional Similarity of Gene Products Based on Gene Ontology. *BMC Bioinformatics*, 7(1): 302, 2006.

[124] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.

[125] B. Schwikowski, P. Uetz, and S. Fields. A Network of Protein-protein Interactions in Yeast. *Nature Biotechnology*, 18(12):1257–1261, 2000.

[126] E. Sefer and C. Kingsford. Metric Labeling and Semi-metric Embedding for Protein Annotation Prediction. In *Proceedings of the 15th Annual International Conference on Research in Computational Molecular Biology*, RECOMB'11, pages 392–407, Berlin, Heidelberg, 2011. Springer-Verlag.

[127] E. Sefer and C. Kingsford. Diffusion Archaeology for Diffusion Progression History

Reconstruction. In *Proceedings of the 2014 IEEE 14th International Conference on Data Mining*, pages 530–539, 2014.

[128] E. Sefer and C. Kingsford. Convex Risk Minimization To Infer Networks From Probabilistic Diffusion Data At Multiple Scales. In *Data Engineering (ICDE), 2015 IEEE 31th International Conference on*, 2015.

[129] E. Sefer, G. Duggal, and C. Kingsford. Deconvolution Of Ensemble Chromatin Interaction Data Reveals The Latent Mixing Structures In Cell Subpopulations. In *Proceedings of the 19th Annual International Conference on Research in Computational Molecular Biology*, RECOMB'15. Springer-Verlag, 2015.

[130] G. Serazzi and S. Zanero. Computer Virus Propagation Models. In *Performance Tools and Applications to Networked Systems*, volume 2965 of *Lecture Notes in Computer Science*, pages 26–50. Springer Berlin Heidelberg, 2004.

[131] B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[132] D. Shah and T. Zaman. Finding Rumor Sources on Random Graphs. *arXiv preprint arXiv:1110.6230*, 2011.

[133] S. Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[134] R. Sharan, I. Ulitsky, and R. Shamir. Network-based Prediction of Protein Function. *Molecular Systems Biology*, 3(1):88, 2007.

[135] D. Sheldon, T. Sun, A. Kumar, and T. Dietterich. Approximate Inference in Collective Graphical Models. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1004–1012, 2013.

[136] Y. Shen, F. Yue, D. F. McCleary, Z. Ye, L. Edsall, S. Kuan, U. Wagner, J. Dixon, L. Lee, V. V. Lobanenkov, and B. Ren. A Map of the Cis-regulatory Sequences in the Mouse Genome. *Nature*, 488(7409):116–120, 2012.

[137] M. Simonis, P. Klous, E. Splinter, Y. Moshkin, R. Willemsen, E. de Wit, B. van Steensel, and W. de Laat. Nuclear Organization of Active and Inactive Chromatin Domains Uncovered by Chromosome Conformation Capture-on-chip (4C). *Nature Genetics*, 38(11): 1348–1354, 2006.

[138] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: A General Repository for Interaction Datasets. *Nucleic Acids Research*, 34(suppl 1):D535–D539, 2006.

[139] T. Stellberger, R. Hauser, A. Baiker, V. Pothineni, J. Haas, and P. Uetz. Improving the Yeast Two-hybrid System with Permutated Fusions Proteins: The Varicella Zoster Virus Interactome. *Proteome Science*, 8(1):8, 2010.

[140] Z. Svitkina and L. Fleischer. Submodular Approximation: Sampling-based Algorithms and Lower Bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

[141] H. Tanizawa, O. Iwasaki, A. Tanaka, J. R. Capizzi, P. Wickramasinghe, M. Lee, Z. Fu, and K.-i. Noma. Mapping of Long-range Associations Throughout the Fission Yeast

Genome Reveals Global Genome Organization Linked to Transcriptional Regulation. *Nucleic Acids Research*, 38(22):8164–8177, 2010.

[142] B. Tolhuis, R.-J. Palstra, E. Splinter, F. Grosveld, and W. de Laat. Looping and Interaction between Hypersensitive Sites in the Active β-globin Locus. *Molecular Cell*, 10(6):1453–1465, 2002.

[143] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving Semidefinite-quadratic-linear Programs Using SDPT3. *Mathematical Programming*, 95:189–217, 2003.

[144] P. Uetz, L. Giot, G. Cagney, et al. A Comprehensive Analysis of Protein-protein Interactions in *Saccharomyces Cerevisiae*. *Nature*, 403(6770):623–627, 2000.

[145] CPLEX. ILOG CPLEX, 2010. http://www.ibm.com/software/integration/optimization/cplex-optimizer.

[146] N. Varoquaux, F. Ay, W. S. Noble, and J.-P. Vert. A Statistical Approach for Inferring the 3D Structure of the Genome. *Bioinformatics*, 30(12):i26–i33, 2014.

[147] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[148] A. Vázquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of Protein Interaction Networks. *Complexus*, 1(1):38–44, 2003.

[149] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Global Protein Function Prediction from Protein-protein Interaction Networks. *Nature Biotechnology*, 21(6):697–700, 2003.

[150] S. Verdú. Computational Complexity of Optimum Multiuser Detection. *Algorithmica*, 4 (1-4):303–312, 1989.

[151] A. Vinayagam, J. Zirin, C. Roesel, Y. Hu, B. Yilmazel, A. A. Samsonova, R. A. Neumller, S. E. Mohr, and N. Perrimon. Integrating Protein-protein Interaction Networks with Phenotypes Reveals Signs of Interactions. *Nature Methods*, 11(1):94–99, 2014.

[152] E. Vynnycky and R. White. *An Introduction to Infectious Disease Modelling*. Oxford University Press, USA, first edition, July 2010.

[153] J. Wallinga and P. Teunis. Different Epidemic Curves for Severe Acute Respiratory Syndrome Reveal Similar Impacts of Control Measures. *American Journal of Epidemiology*, 160(6):509–516, 2004.

[154] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[155] L. A. Wolsey and G. L. Nemhauser. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1999.

[156] E. Yaffe and A. Tanay. Probabilistic Modeling of Hi-C Contact Maps Eliminates Systematic Biases to Characterize Global Chromosomal Architecture. *Nature Genetics*, 43(11): 1059–1065, 2011.

[157] J. Yang and J. Leskovec. Patterns of Temporal Variation in Online Media. WSDM '11, pages 177–186, New York, NY, USA, 2011. ACM.

[158] Y. Zhang, R. McCord, Y.-J. Ho, B. Lajoie, D. Hildebrand, A. Simon, M. Becker, F. Alt, and J. Dekker. Spatial Organization of the Mouse Genome and Its Role in Recurrent Chromosomal Translocations. *Cell*, 148(5):908 – 921, 2012.

[159] Z. Zhang, G. Li, K.-C. Toh, and W.-K. Sung. Inference of Spatial Organizations of Chromosomes Using Semi-definite Embedding Approach and Hi-C Data. In *Research in Computational Molecular Biology*, volume 7821 of *Lecture Notes in Computer Science*, pages 317–332. Springer Berlin Heidelberg, 2013.

[160] J. D. Ziebarth, A. Bhattacharya, and Y. Cui. CTCFBSDB 2.0: A Database for CTCF-binding Sites and Genome Organization. *Nucleic Acids Research*, 41(D1):D188–D194, 2013.