Imperial College London
Department of Computing

Methods for Analysis and Integration of Heterogeneous Network Data

Vladimir Gligorijević

June 2017

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London and
the Diploma of Imperial College London.

Declaration

I herewith certify that the original work submitted in this dissertation is my own and that all material in this dissertation which is not my own work has been properly acknowledged.

Vladimir Gligorijević

Copyright

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Abstract

In many areas of science and technology data describing a phenomenon or a system of interest and its individual components can be obtained from different types of instruments, using different acquisition techniques and experimental setups. Furthermore, many systems can be characterized by different data sets obtained by observing the system from different perspectives, under different conditions, or at different points in time. For example, in biology and biomedicine, rapid technological advances have led to the production of such *heterogeneous* and *multimodal* data, and enabled construction of complex networks with various types of interactions between diverse biological entities. Conventional network data analysis methods were shown to be limited in dealing with such multi-relational network data, as they are designed specifically for analysis of single-relational networks.

Therefore, in this Thesis, we propose novel integrative methods that can collectively mine multiple types of biological networks and create more accurate integrative models capable of producing more holistic, systems-level biological insights. The proposed methods are based on Non-negative Matrix Factorization, a dimensionality reduction technique, that is further extended for learning and modelling dependencies between large-scale heterogeneous and multimodal network data. Our methods have a wide range of applications of which we focus on those from systems biology and biomedicine, including *protein function prediction*, *multiple network alignment*, *extraction of composite functional modules from multimodal biological networks*, and problems in *precision medicine*, including *patient stratification*, *drug repurposing* and *cancer driver gene prediction*. In each of these applications we show that our methods outperform existing methods, provide biological insights and generate novel hypothesis that could guide future research and biological experiments.

Acknowledgements

I would like to acknowledge several individuals who played significant roles through the course of my Ph.D. studies.

First of all, I would like to thank my supervisor Dr. Stefanos Zafeiriou who taught me the art of machine learning and provided me with wise guidance whenever I needed. The skills and knowledge which I have gained in numerous discussions with Dr. Zafeiriou were not only instrumental for the completion of this thesis but also a valuable component in my future career development. I would also like to thank my second supervisor, Dr. Michael Sternberg for his helpful mentoring and for his patience in proof-reading this thesis. I was always warmly welcomed to participate in his group meetings and I truly enjoyed numerous discussions with his group members.

I would also like to thank my ex-supervisor Dr. Nataša Pržulj, who entirely changed my career path from a physicist to a computer scientists by showing me an exciting world of bioinformatics. She engaged me in numerous challenging and interesting projects that allowed a lot of innovative thinking and provided me with skills required for being an independent researcher.

In addition to my supervisors, I would like to thank my collaborators and colleagues, Dr. Noël Malod-Dogning, Dr. Anida Sarajlić and Dr. Yannis Panagakis for their help and insightful comments that shaped the work presented in this thesis.

I am especially grateful to my best friends, Nemanja Rakićević, Robert Walecki and Nikola Ojkić for sharing numerous lunch breaks and discovering vast range of beers and pubs in London that helped me overcome difficult moments in my Ph.D.

I would also like to acknowledge Department of Computing at Imperial College London for their financial support. My research has also been supported by Dr. Przulj's ERC Starting Independent Researcher Grant 278212.

Finally, I would like to thank my wife, Natalija Odanović, for her constant support and love through my Ph.D. studies.

This dissertation is dedicated to my parents, Goran and Živanka Gligorijević, for their constant love and support throughout my entire education.

Contents

Abstract **3**

Acknowledgements **4**

1 Introduction **14**

1.1 Motivation and objectives 14

1.2 The need for data integration in systems biology and biomedicine . . 16

1.3 Thesis contributions and overview 18

1.4 List of publications 21

2 Background **22**

2.1 Network representation and basic network properties 22

2.2 Review of biological data types and molecular networks 25

2.3 Review of machine learning methods for biological data integration . 28

2.3.1 Bayesian data integration 29

2.3.2 Kernel-based data integration 31

2.3.3 Matrix factorization-based data integration 34

3 Integration and analysis of heterogeneous multi-relational networks **39**

3.1 Network integration for protein function prediction and GO reconstruction 40

3.1.1 Introduction 40

3.1.2 Methods 42
3.1.3 Integration of various molecular networks 44
3.1.4 Multiplicative update algorithm 46
3.1.5 Predicting associations between GO terms 47
3.1.6 Rank parameters selection 48
3.1.7 Data sets and preprocessing 49
3.1.8 Results and discussion 50
3.1.9 Conclusions 54
3.2 Data integration for precision medicine 55
3.2.1 Introduction 55
3.2.2 Methods 57
3.2.3 Datasets, preprocessing and matrix construction 62
3.2.4 Results 63
3.2.5 Conclusions 68
3.3 Multiple Network Alignment via network and sequence integration . . 71
3.3.1 Introduction 71
3.3.2 Methods 74
3.3.3 Results and Discussion 80
3.3.4 Conclusions 87

4 Integration and analysis of homogeneous multi-relational networks 88

4.1 Introduction: from single-layer to multilayer networks 89
4.1.1 Single-layer (monoplex) networks 90
4.1.2 Multilayer networks 92
4.2 Clustering evaluation measures 94
4.3 Decomposition of network layers into a common non-negative representation 96

4.3.1 Weighted Simultaneous SNMTF 97
4.3.2 Optimization with ordinary gradient 98
4.3.3 Optimization with natural gradient 99
4.3.4 Experiments 100
4.3.5 Results and discussion 103
4.3.6 Conclusion 105
4.4 Fusing nonnegative representation of graphs on Grassmann manifold . 106
4.4.1 Model setup 107
4.4.2 Merging low-dimensional representation of graph layers on Grassmann Manifolds 109
4.4.3 Derivation of the general multiplicative update rules 110
4.4.4 Experiments 113
4.4.5 Setup for state-of-the-art methods 117
4.4.6 Results and Discussion 118
4.4.7 Conclusion 122

5 Conclusion **124**
5.1 Summary of Thesis Achievements 124
5.2 Future Work 127
5.2.1 Clustering dynamical multilayer networks 127
5.2.2 Inference in multi-relational networks 132
5.2.3 Gene function prediction from multilayer networks 136

Bibliography **137**

Appendix A Important concepts from molecular biology **163**

Appendix B Reconstruction of GO by integration of molecular networks **165**

B.1 Supplementary figures 165

Appendix C Multiple network alignment **167**

C.1 Study of different parameters 167

C.2 Study of different alignment orderings 171

C.3 Permutation test for determining p-values of predicted associations . 171

C.4 Transferring annotations across species 172

Appendix D Fusing network layers on Grassmann manifold **174**

D.1 Artificial Multiplex Networks 174

D.2 Derivation of update rules for NF-CCE algorithm 174

D.3 Parameter estimation 180

List of Tables

2.1 Summary of biological data and network types. 25

3.1 Network used in the protein function predicitno and GO reconstruction study. 49

3.2 The list of the top scoring proposed ovarian cancer driver genes. . . . 67

3.3 Official gene names of the predicted ovarian cancer driver genes. . . . 68

3.4 The list of predicted top scoring drug-target associations. 69

3.5 The five PPI networks considered in the Multiple Network Alignment study. 76

3.6 Functional consistency analysis. 85

4.1 Clustering accuracy measures for three synthetically generated multi-layer networks, *SYNT-1*, *SYNT-2* and *SYNT-MIX*. 104

4.2 Clustering accuracy measures for three real world multi-layer networks: CiteSeer, CoRA and MIT. 105

4.3 Multiplicative update rules (MUR) for single-layer and multilayer network analysis. 111

4.4 Real-world multilayer networks used for our comparative study. . . . 117

4.5 Clustering accuracy of MM, SNMF, PNMF, SNMTF, SsNMTF, PMM, SNF, SC-ML, LMF, GraphFuse, CGC-NMF, CSNMF, CPNMF, CSNMTF, CSsNMTF applied on real-world multilayer networks. 120

D.1 Optimal parameters for PMM, SC-ML, LMF, GraphFuse, CSNMF, CPNMF, CSNMTF, CSsemi-NMTF applied on real-world multi-layer networks. 180

List of Figures

1.1 An illustration of "omics layers". 15

1.2 The roadmap of the topics that will be covered in this Thesis. 20

2.1 Two types of multi-relational networks: *homogeneous* and *heterogeneous* multi-relational network. 24

2.2 Drug clustering from heterogeneous network data in the setup of KB-based and MF-based integration methods. 32

3.1 Schematic representation of multi-relational network used in protein function prediction and GO reconstruction study. 46

3.2 Relative contribution of each data source to the integration model measured by residual sum of squares (RSS) and explained variance (Evar). 51

3.3 (a) The fraction of reproduced GO terms in each category, i.e., CC, BP and MF. (b) Validation of predicted GO terms by new genetic interaction profiles data. 53

3.4 The top 25 most frequently mutated genes in TCGA serous ovarian cancer patients. 56

3.5 Patient-specific data integration framework. 58

3.6 Schematic illustration of data sets used in the patient-specific data integration study. 59

3.7 Kaplan-Meier survival curves for 3 different patient groups produces by GNMTF (panel A) and NBS (panel B). 64

3.8 Enrichment of identified gene clusters in mutated and known ovarian cancer drivers. 65

3.9 Area under the ROC and PR curves for GNMTF runs done on the combination of particular input data sets listed on x-axis. 70

3.10 An illustration of the basic principle of GNMTF-based data integration of 5 PPI networks. 78

3.11 Functional consistency of GNMTF associations. 81

3.12 Coverage analysis in terms of number of clusters and number of proteins. 83

3.13 Functional consistency analysis. 84

4.1 An example of multilayer networks with 11 nodes present in three complementary layers denoted in different colors. 90

4.2 Spy-plots of three random synthetic multilayer networks with three layers. 101

4.3 (top) CiteSeer and (bottom) CoRA multi-layer networks. 104

4.4 The clustering performance of our proposed and other methods on 11 different *SYNTH-C* multilayer networks measured by NMI. 118

4.5 The clustering performance of our proposed and other methods on 10 different *SYNTH-N* multilayer networks measured by NMI. 119

4.6 Average redundancy of each individual network layer (in blue), computed by *SNMF*, and of their fused representation (in red), computed by *CSNMF*, for networks: (A) HBN, (B) YBN and (C) MBN. Method parameters are: $k = 200$ and $\alpha = 0.01$. 122

4.7 Average number of GO terms significantly enriched (i.e., p-val $\leq$ 0.05) in clusters of each individual network layer (in blue), computed by *SNMF*, and of their fused representation (in red), computed by *CSNMF*, for networks: (A) HBN, (B) YBN and (C) MBN. Method parameters are: $k = 200$ and $\alpha = 0.01$. 123

5.1 An illustration of dynamical multilayer network. 128

5.2 Two types of inference in multi-relational networks. 133

5.3 ROC curves representing the performance of (A) *DFMF* and (B) *NIMF* in reconstructing links removed from DTI network. 135

5.4 Schematic representation of *NF-CCE* framework repurposed for gene function prediction. 137

B.1 (A) An illustration of graphlets and orbits; (B) An example of graph and GDV vector; (C) An example of adjacency and Laplacian matrices for a given graph. 165

B.2 GDV similarity distributions in four different yeast networks: PPI, GI, Co–Ex and YeastNet. 166

B.3 Estimation of rank parameters by computing cophenetic correlation coefficient. 166

C.1 Functional consistency of GNMTF and sequence–based associations for different values of γ. 167

C.2 Functional consistency of Fuse's clusters for different values of γ. . . . 168

C.3 Functional consistency of Fuse's clusters for different values of α. . . . 169

C.4 2–fold cross validation for estimating α parameter. 170

C.5 Functional consistency of Fuse's clusters when using different alignment step orderings. 171

C.6 Cluster BP (left panel) and MF (right panel) consistency of all GNMTF–predicted protein associations. 172

D.1 Spy–plots of five different 2–layer artificial multiplex networks used in our experiments to demonstrate the effect of *noise* on the performance of our methods. 174

D.2 Spy–plots of five different 2–layer artificial multiplex used in our experiments to demonstrate the ability of our methods to integrate complementary layers. 175

Chapter 1

Introduction

1.1 Motivation and objectives

In many areas of science and technology data describing a phenomenon or a system of interest and its individual components can be obtained from different types of instruments, using different acquisition techniques and experimental setups. Also, we may collect different data sets that are directly describing the system, or data sets that are indirectly related to the system, but when combined with other data sets they could provide very informative insights into the system. Thus, various data sets are often *heterogeneous*. Mining heterogeneous data sets introduces new challenges and questions that are beyond those related to mining each data set separately. For example, when classifying patients into different categories of disease severity, patients' clinical data together with their *exposomics* data (demographics, habits, living environments, etc.) represent a valuable complement to the patients' genetic data. Collective analysis of all data types results in a more accurate classification than the classification obtained when a single data type is considered.

Furthermore, many systems can be characterized by different data sets obtained by observing the system from different perspectives, under different conditions, or at different points in time. Thus, many data sets are dynamical (i.e., *time-dependent* data) or characterized with multiple modalities (i.e., *multimodal* data). One such example of a system, to which we will pay more attention in this Thesis, is the *human cell*. The human cell is one of the most studied complex systems. However, its functioning is still largely unknown. It comprises diverse molecular structures, forming complex, dynamical molecular machinery, which can be naturally represented as a system of various types of interconnected molecular and functional networks (see Fig. 1.1 for an illustration). Recent technological advances in high-throughput biology have generated vast amounts of disparate biological data describing different

aspects of cellular functioning also known as *omics* layers. For example, yeast two-hybrid (Y2H) assays [1–5] and affinity purification with mass spectrometry (AP/MS) [6, 7] are the most widely used high-throughput methods for identifying physical interactions (bindings) among proteins. These interactions, along with the whole set of proteins, comprise the *proteome* layer. Other experimental technologies, such as Next Generation Sequencing (NGS) [8, 9], have enabled construction and analyses of other *omics* layers. Figure 1.1 illustrates these layers and their constituents: genes in the *genome*, mRNA in the *transcriptome*, proteins in the *proteome*, metabolites in the *metabolome* and phenotypes in the *phenome*. The figure illustrates one important fact related to why collective analysis of all layers is important. Namely, if we want to understand a mechanism by which genes (in the genome layer) lead to complex phenotypes (in the phenome layer), we must take into account all intermediate layers and their mutual relationships (e.g., protein-DNA interactions).

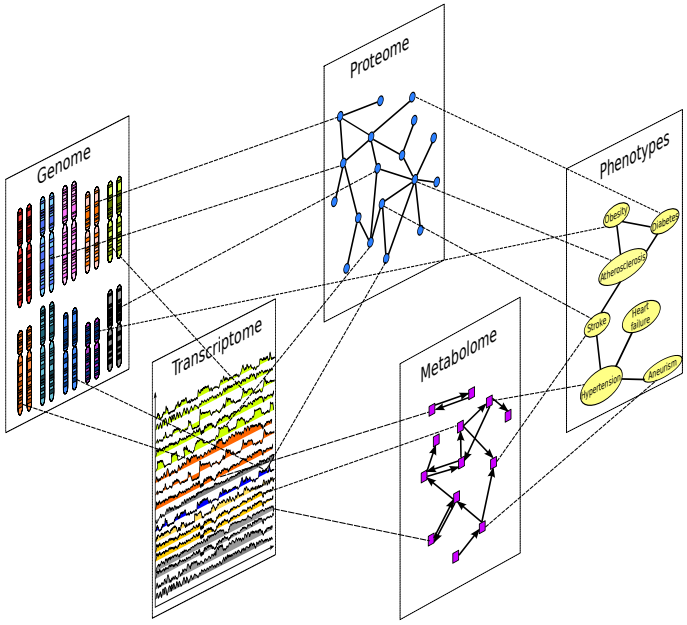


Figure 1.1: An illustration of the representation of a cell's molecular organization. The representation includes four *omics* layers each of which could be represented by a molecular network. Interactions between components in different layers are represented by dashed lines. The figure is taken from Gligorijević and Pržulj [10].

It has largely been accepted that a comprehensive understanding of a biological system [1] can only come from a joint analysis of all its parts (i.e., *omisc layers* [11, 12]. Such analysis is often referred as *data fusion (or data integration)*. Data integration collectively analyses all data sets and builds a joint, predictive model that captures all data sets concurrently.

In this Thesis, we will be mostly focusing on the biological networked data sets. Thus, a starting point in a data integration analysis is to use a mathematical concept

[1] henceforth the term *biological system* will either refer to a cell or a patient

of *networks* to represents *omics* layers. A network (or a *graph*) consists of *nodes* (or *vertices*) and *links* (or *edges*). In biological networks, nodes usually represent discrete biological entities at a molecular (e.g., genes, proteins, metabolites, drugs etc.) or phenotypic level (e.g., diseases), whereas, edges represent physical, functional, or chemical relationships between pairs of entities [13] (for a detailed summary of types and classification of biological networks considered in this Thesis, see Section 2.2). Recent advances in biotechnology have led to previously unseen rates of growth in the number of different types of biological entities (e.g. genes, RNAs, proteins, metabolites, drugs, diseases, etc.) captured with different technologies having varying coverage, bias, and noise robustness [10]. These data sets have provided an opportunity for construction of *heterogeneous, multi-relational* networks capable of capturing more comprehensively different types of biological entities and their mutual relations (see Figure 1.1 and Section 2.2 for details) [14]. Standard network data analysis methods were shown to be limited in dealing with such heterogeneous networked data. Hence, new methods that can cope with these complexities and that can collectively mine multiple types of biological data and produce more holistic, systems-level biological insights are needed.

1.2 The need for data integration in systems biology and biomedicine

Abundance of different types of biological data has made data integration approaches increasingly powerful and popular over the past decade. Understanding cellular processes and molecular interactions by integrating *omics* layers is one of the challenges in systems biology. Some of the other challenges, that could benefit from integrative approaches, include *biological network inference*, *protein function prediction*, *drug repurposing* and challenges related to *precision medicine*, such as *patient stratification* (or *disease subtyping*), *biomarker discovery* and *personalized treatment prediction*. Here we briefly summarise these biological challenges and point out to the importance of integrative approaches in addressing them. These challenges will be addressed by our methods in more details in the rest of this Thesis. For definitions of biological terms used in this Thesis, please refer to glossary in Appendix A.

Biological network inference. Network inference is one of the major problems in systems biology aiming to understand gene interactions and their mutual influence [15, 16]. It aims to construct network topology (or wiring between genes or other biological entities) based on the information contained in different data types. The special focus in the literature has been on the inference of gene regulatory networks (GRNs), whose nodes are genes and edges are regulatory interactions[17]. Standard

methods for GRN inference have mostly been based on gene expression data. However, integration of expression data with other data types, such as genome sequence or protein-DNA interaction data, has been shown to improve the GRN inference process [16]. In addition to GRN, functional linkage (or association) network (FLN) is another type of network that is often referred to in the literature. FLN captures all possible gene associations, constructed from multiple data types. FLN has demonstrated its potential in studying human diseases, postulating for new disease causing genes and predicting novel gene functions and interactions [18, 19].

Protein function prediction. A protein function is a broad term and it usually represents a set of roles of any particular protein inside an organism [20]. The Gene Ontology (GO) represents a systematic classification of all known protein functions organized as well-defined terms (also known as GO terms) divided into three main categories, namely Molecular Function (MF), Biological Process (BP) and Cellular Component (CC) [21]. Protein function prediction has demonstrated to be a good alternative to the time consuming experimental protein function characterisation. It aims to computationally assign functions to unannotated or incompletely annotated proteins. The accuracy of these methods has largely improved with the use of integration methods that can incorporate multiple different biological data representing complementary information about protein functions [22–26]. For example, a single protein can be characterized by its amino acid sequence, a set of its interacting neighbors in protein-protein interaction network, or the expression level of its corresponding gene. These bits of information represent valuable complements to each other and can be jointly utilized to make a more accurate model of protein function prediction. Furthermore, interactions between proteins are important for understanding intracellular signalling pathways and other cellular process. However, structure of PPI networks and protein functions for many species are still largely unknown and therefore, many computational techniques for PPI interaction prediction and transfer of annotations from one specie to another have been proposed. Much attention has been paid to the data integration methods capable of inferring new PPIs by integrating various types of heterogeneous biological data [19, 27, 28].

Drug repurposing. It aims to find new uses for existing drugs and thereby drastically reduce the cost and time for new drug discovery [29]. Accumulation of various biological data involving interactions between drugs, diseases and genes, and protein structural and functional similarities, provide new opportunities for data integration methods to generate new associations between diseases and existing drugs [30].

Precision medicine. It deals with the identification of risk factors and adaptation of management and treatment to individual patients [31]. Unlike traditional diagnosis and treatment, personalized medicine hopes to offer better, data-driven

diagnoses. The major problem of precision medicine is the molecular diversity of patients even with the same clinical phenotypes. Thus, integrative approaches that utilize patient-specific molecular genomic information as well as clinical data (e.g., patient history, laboratory analysis, ultrasound parameters, etc.) are expected to overcome this problem. Data integration methods have started contributing to this growing field [32]. Such a data-driven diagnosis could lead to earlier intervention and more efficient drug deployment and treatment [33]. Some of the challenges of precision medicine include, *patient stratification* (or *disease subtyping*), i.e., grouping patients into subgroups with different clinical outcomes [34], *biomarker discovery*, i.e., detecting any measurable diagnostic indicator that is used to assess the risk, or presence of a disease [35] and *personalized treatment prediction* [36].

Although different, these challenges have the same general approach used to address them - integrative approach. The approach is general and it is reflected in the fact that each individual data set, describing one aspect of the problem at study, often contains complementary or incomplete information. Thus, the model built on all data sets is believed to be more reliable and accurate than the model build on any single data set alone. This stresses the need for integrative methods in systems biology and biomedicine.

1.3 Thesis contributions and overview

This Thesis introduces a new class of data integration methods for mining *homogeneous and heterogeneous multi-relational* and *time-dependent* networked data. The major applications of these methods will be demonstrated on the biological problems introduced in the previous section and presented in Chapter 3; however, in Chapter 4 we will also introduce other types of systems for which integrative analysis is also beneficial. The contribution of the Thesis can be divided into two parts:

Methodological contributions: Development of novel machine learning methods for mining and integration of *homogeneous and heterogeneous multi-relational* and *time-dependent* networked data.

- Novel Matrix Factorization-based data integration methods for mining *heterogeneous multi-relational networks*, i.e., networks with multiple types of nodes and edges (see Fig. 2.1, left). The methods are designed to address the following tasks: *heterogeneous link inference*, *co-clustering of nodes of different types* and *inference of networks with homogeneous and heterogeneous links*.
- Novel Matrix Facotirization-based data integration methods for mining *homogeneous multi-relational networks*, i.e., networks with different types of links

connecting the same type of nodes (see Fig. 2.1, right). The methods are designed to address the followings tasks: *composite community detection* and *integrated network inference*.

Biological contributions: The developed methods were applied to real-world biomedical problems, introduced in the previous section, providing the following results:

- Prediction of novel functions of partially annotated yeast's proteins by integrating existing annotations with various types of yeast molecular networks; and prediction of novel hierarchical associations between the functions (GO terms) themselves. Inference of GO hierarchical associations solely from molecular networks.
- Comparison of different species by integrating protein sequence and protein-protein interaction networks; and prediction of proteins pairs that are evolutionary and functionally conserved across species, and that could be used for transfer of annotations from well-observed proteins, in one species, to partially observed proteins in another species.
- Identifying subgroups of ovarian cancer patients having different clinical outcome by integrating somatic mutation data, molecular networks and drug data; predicting novel driver genes responsible for the ovarian cancer progression; and predicting novel drug candidates for repurposing, that could be used in the treatment of ovarian cancer patients.
- Finding composite clusters of functionally consistent genes in a collection of molecular interaction networks.

This Thesis will first provide the reader with a background and basic concepts from network science, matrix factorization and molecular biology. Then it will provide a reader with a review of existing methods for data integration. Finally, it will present new integrative analysis methods and techniques and demonstrate their application in addressing the biological problems introduced in the previous section. Therefore, the Thesis is grouped into the following chapters (see Fig. 1.2 for an illustration):

Chapter 2 describes basic concepts and definitions for understanding this Thesis. The chapter will provide two types of reviews: a review of types and classes of biological networks and data sets and repositories used for their construction; and finally, a literature review about current *state-of-the-art* machine learning techniques

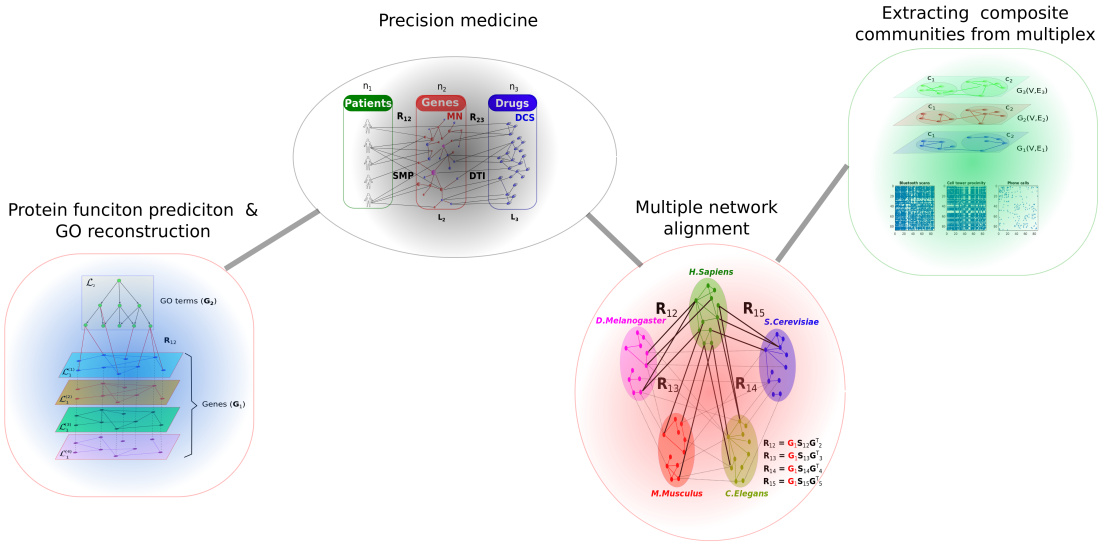


Figure 1.2: The roadmap of the topics that will be covered in this Thesis.

and methods for data analysis and integration, with a special focus on existing matrix factorization techniques, based on which we will propose novel methods.

Chapter 3 proposes an existing approach, namely *Non-negative Matrix Tri-Factorizaton*, with novel modifications introduced in this chapter, as a new general framework for integration of biological data represented as *heterogeneous multi-relational networks*. The framework is applied on heterogeneous biological data in three seemingly unrelated biological problems: 1) protein function prediction and GO inference (Section 3.1); 2) multiple network alignment of protein-protein interaction networks (Section 3.2); and 3) addressing precision medicine challenges, namely patient stratification, driver gene prediction and drug repurposing, for ovarian cancer patients (Section 3.3).

Chapter 4 introduces a novel class of *Non-negative Matrix Factorization* methods for mining *homogeneous multi-relational networks*. The methods utilize all link types (network layers) contained in different data sets to construct consensus clusters covering all data sets. The methods are applied to various types of multimodal (multiplex) networks, ranging from biological networks to social and world trade networks, and their performance is compared to existing *state-of-the-art* methods.

Chapter 5 will provide a concluding summary and general discussion of the novel methods and results presented in the Thesis, as well as an outlook on possible future work.

1.4 List of publications

Major results of this thesis have been previously published in several peer-reviewed scientific journals and conferences, or prepared as articles which are currently under review for publication. Below is the list of original articles. The text and figures taken from the papers, together with the contributions of co-authors are properly acknowledged and identified in the relevant Chapters.

1. Gligorijević, V., Janjić, V. & Pržulj, N. (2014). Integration of molecular network data reconstructs Gene Ontology. *Bioinformatics*, 30(17), i594-i600.

2. Gligorijević, V., Malod-Dognin, N. & Pržulj, N. (2016). Patient-specific data fusion for cancer stratification and personalised treatment. *Biocomputing, World Scientific, 2016*, 321-332.

3. Gligorijević, V., Malod-Dognin, N. & Pržulj, N. (2015). Fuse: multiple network alignment via data fusion. *Bioinformatics*, 32(8), 1195-1203.

4. Gligorijević, V. & Pržulj, N. (2015). Methods for biological data integration: perspectives and challenges. *Journal of The Royal Society Interface*, 12(112), 20150571.

5. Gligorijević, V., Malod-Dognin, N. & Pržulj, N. (2016). Integrative methods for analyzing big data in precision medicine. *Proteomics*, 16(5), 741-758.

6. Gligorijević, V. & Pržulj, N. (2016). Computational methods for integration of biological data. *In Personalized Medicine* (pp. 137-178). Springer International Publishing.

7. Gligorijević, V., Panagakis, Y. & Zafeiriou, S. (2016). Fusion and Community Detection in Multi-layer Graphs. *Pattern Recognition (ICPR), 23rd International Conference on, 1327-1332.*

8. Gligorijević, V., Panagakis, Y. & Zafeiriou, S. (2017). Non-Negative Matrix Factorizations for Multiplex Network Analysis. *arXiv:1612.00750.*

Chapter 2

Background

In this chapter, first, we cover important concepts from network (graph) theory and introduce two important network representations that will be extensively used in the rest of this thesis: *homogeneous* and *heterogeneous multi-relational* network representations (Section 2.1). Then, we present two literature reviews: 1) a review of different types of networks and available biological data commonly used in the their construction, presented in Section 2.2; and 2) a review of general, machine learning approaches to biological data integration, including *bayesian*, *kernel-based* and *matrix factorizaton* methods, presented in Section 2.3. Concepts and some parts of this chapter have previously been published in: Gligorijević and Pržulj [37] and Gligorijević and Pržulj [10].

2.1 Network representation and basic network properties

Biological networks have revolutionized our approach to studying biological systems and disease, as they enabled a shift from a single-gene to a system-level perspective. A network (or a graph), usually denoted as $G = (V, E)$, consist of a set of nodes, V, of size $n = |V|$, and a set of edges, E, of size $m = |E|$ [38]. Depending on a type of data they represent, network edges can be *directed* or *undirected*, *weighted* or *unweighted* [39]. For example, an edge in a PPI network is undirected, as it represents a physical binding between two proteins, whereas an edge in a metabolic network is directed, as it represents a one-way chemical reaction that converts one metabolite into another. A network connectivity pattern is often represented by an *adjacency matrix* [38]: for an undirected and unweighted network $G = (V, E)$, its adjacency matrix, $\mathbf{A}$, is a square matrix of size $n \times n$, where each row and column

denotes a node and entries in the matrix are either $\mathbf{A}_{ij} = 1$, if nodes $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, n\}$ are connected, or $\mathbf{A}_{ij} = 0$, otherwise. In the case of a weighted network, instead of binary values, entries in an adjacency matrix are real numbers representing the strengths of associations between the nodes. The *Laplacian matrix* of G, denoted as $\mathbf{L}$, is another mathematical concept widely used in spectral graph theory [40] and machine learning problems [41]. It is defined as: $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the diagonal degree matrix; that is, entries on the diagonal, $\mathbf{D}_{ii}$, where $i \in \{1, \ldots, n\}$, are node degrees (the degree is the number of edges connected to a node), whereas off-diagonal entries, $\mathbf{D}_{ij}, i \neq j$, are zeros.

Networks can also be used to model relations between different types of biological entities, such as genes and diseases. Such relations are usually represented by using *bipartite* networks. Namely, a bipartite (or in a more general case, a k-partite) network, $G = (V_1, V_2, E)$, consists of two (or k) disjoint sets of nodes (partitions), V_1 and V_2, and a set of edges, E, connecting nodes between different partitions [38]. For example, gene-disease associations could be represented by a bipartite network. A connectivity pattern of an unweighted, undirected bipartite network can be represented by an *adjacency matrix*, $\mathbf{R}$, of size $|V_1| \times |V_2|$, where each row, $i \in \{1, \ldots, |V_1|\}$, denotes a node in V_1 and each column, $j \in \{1, \ldots, |V_2|\}$, denotes a node in V_2, and entries in the matrix are either $\mathbf{R}_{ij} = 1$, if nodes $i \in V_1$ and $j \in V_2$ are connected, or $\mathbf{R}_{ij} = 0$, otherwise.

Network wiring (also called *topology*) has intensively been studied over the past couple of decades by using methods from graph theory and statistical physics [39]. The first studies of molecular networks have shown that many molecular networks are characterised by a *complex, scale-free structure*. Namely, they have a small number of highly connected nodes (hubs) whose removal disconnects the network and a large number of low-degree nodes [42]. Unlike the structure of *random networks*, such scale-free structures indicate that molecular networks emerge as a result of complex, dynamical processes taking place inside a cell. This property has been exploited in devising null models of these networks [43–45].

Combinations of unipartite and bipartite network representations are usually used to link multiple types of networks into a single, complex, multi-relational network. Such a representation is more suitable for describing complex, heterogeneous biological data comprising of different types of entities. Here, we distinguish between two types of multi-relational networks[1]:

[1] note that in the literature, different research communities use different terms to denote these networks [46]. Formally, in the graph theory community, these networks are known as *edge-labeled multigraphs* and *multipartite edge-labeled multigraphs*, respectively. However, a standardized naming convention is still lacking. Some of the frequently used terms include *multidimensional, multimodal, multilayer, multiplex, multilevel, interconnected, etc.*

- *heterogeneous multi-relational networks*, $G(\bigcup_{i=1}^{M} G_i, E_R)$, representing a collection of L interconnected network layers, $G_i = (V_i, E_i)$, each of which having a different type of vertices, V_i, and intra-layer edges, E_i, connecting them; and a set of inter-layer edges, E_R, connecting vertices of different network layers. See Fig. 2.1 (left panel) for an illustration.

- *homogeneous multi-relational networks*, $G(V, \bigcup_{i=1}^{M} E_i)$, representing a collection of M networks (also referred as layers), $G_i(V, E_i)$, having in common a set of vertices, V, whereas a set of intra-layer edges, E_i, varies across network layers. See Fig. 2.1 (right panel) for an illustration.

For example, in many network integration studies, genes in a gene-gene interaction network and diseases in a disease-disease association network are linked by a gene-disease association bipartite network, jointly forming a complex, heterogeneous, multi-relational network [47]. The most general type of multi-relational network is obtained by combining these types of representations. For example, by replacing each layer of the *heterogeneous multi-relational network*, $G_i(V_i, E_i)$ by a set of, M_i layers of *homogneous multi-relational network*, $G_i^{(j)}(V_i, E_i^{(j)})$, where $j \in \{1, \ldots, M_i\}$.

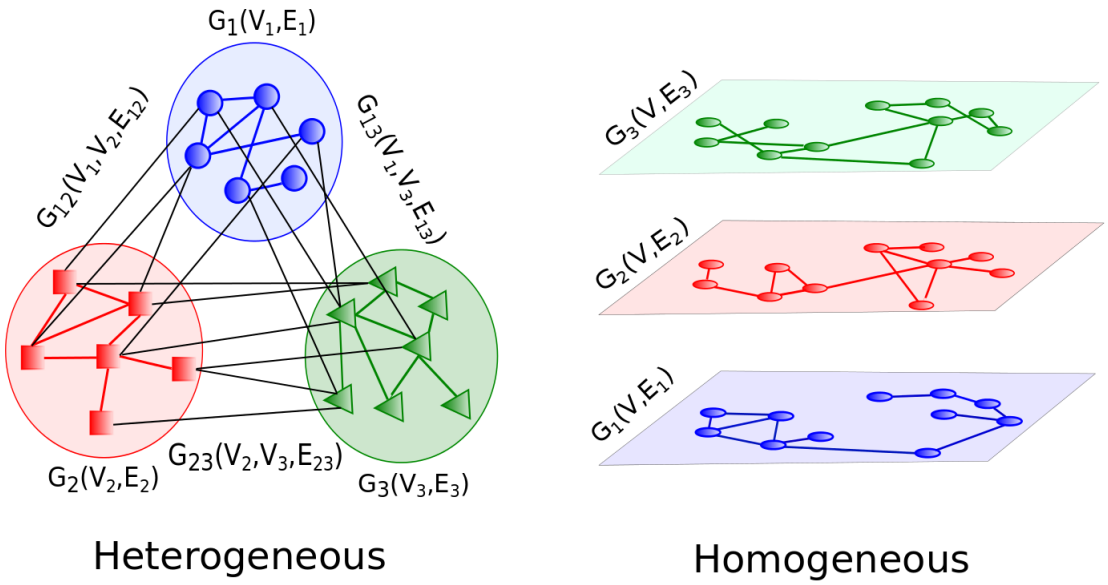


Figure 2.1: *Left:* Heterogeneous multi-relational network consisting of three interconnected networks having three types of nodes represented by blue circles, red squares and green triangles, respectively. *Right:* Homogeneous multi-relational network consisting of three types of layers, each of which having the same type of nodes.

Over the past couple of decades, a variety of mathematical tools for extraction of biological knowledge from real-world molecular networks has been proposed. In

Network type	Network	Entities/Nodes	Interactions/Edges	Data resource
Molecular interactions	PPI	proteins	physical bindings	BioGRID [49]
	MI	enzymes (proteins)	reaction catalyzation	KEGG [50]
	DTI	drugs/targets	physical bindings	DrugBank [51], PubChem [52]
Functional associations	GI	genes (proteins)	genetic interactions	BioGRID [49]
	GDA	genes/diseases	associations	OMIM [53], GWAS [54] PheWAS [55]
	ON	GO (DO) term	hierarchical relations	GO [21], DO [56]
Structural/Functional similarities	Co-Ex	genes	expression correlations	GEO [57], ArrayExpress [58], SMD [59]
	DCS	drugs	structural similarities	DrugBank [51], PubChem [52]
	DSES	drugs	side-effect similarities	SIDER [60]
	PSeqS	proteins	sequence similarities	RefSeq [61]
	PStrS	proteins	structural similarities	PDB [62]

Table 2.1: Table of different types of biological network (the first two columns), types of biological entities and relations (interactions) between them (the second two columns) and some databases containing the data (the last column).

this Thesis, we do not provide a review of these methods, because they are mainly used for single-type network analyses. These methods have recently been reviewed in [48]. Here, we present a brief description of different types of biological networks and review integrative methods commonly used in their analysis.

2.2 Review of biological data types and molecular networks

Based on the criteria under which the links in the networks are constructed, we divide biological networks into the following three categories (see Table 2.1 for a summary of network types and data repositories):

1) Molecular interaction networks. They include the following network data types.

- *A protein-protein interaction (PPI) network* consists of proteins (nodes) and physical bindings between them (undirected edges). A large number of studies have dealt with detection and analysis of these types of interactions in different species [63, 64]. As proteins are coded by genes, a usual way to denote nodes in a PPI network is by using gene notations. Such notations are more common, as they allow a universal representation of all molecular networks and enable their comparison and integration.

- *A metabolic interaction (MI) network* contains all possible biochemical reactions that convert one metabolite into another, as well as regulatory molecules, such as enzymes, that guide these metabolic reactions [65]. A usual way to represent a metabolic network is by representing enzymes as nodes and two enzymes are linked if they catalyze (participate in) the same reaction [66]. Since

enzymes are proteins (i.e., sintesized from the genome), they can be denoted by a gene notation.

- *A drug-target interaction (DTI) network* is a bipartite network representing physical bindings between drug compounds in one partition and target proteins in the other [67]. Many databases containing curated DTIs from scientific literature have been published (see Table 2.1).

2) Functional association networks. They include the following network types:

- *A genetic interaction (GI) network* is a network of genes representing the effects of paired mutations onto the organism's phenotype. Two genes are said to exhibit a *positive (negative) genetic interaction* if their concurrent mutations result in a better (worse) phenotype than expected by mutations of each one of the genes independently [13, 68]. Genetic interactions may not represent physical "interactions" between the proteins, but their functional associations.
- *A gene-disease association (GDA) network* is a bipartite network representing associations between diseases in one partition and disease-causing genes in the other partition [69].
- *Ontology networks (ON)* are valuable biological components in many network integration studies [70] and they are often integrated with other molecular network data. The two commonly used ontologies in network data integration literature are Gene Ontology (GO), which unifies the knowledge about functioning of genes and gene products [21], and Disease Ontology (DO), which unifies the knowledge about relationships between diseases [56]. The hierarchical structure of the ontologies is represented as a Directed Acyclic Graph (DAG) - a graph with no cycles, where nodes represent GO terms (or DO terms) and edges represent parent-child semantic relations.

3) Similarity networks. They include the following network data types:

- *A gene co-expression (Co-Ex) network* represents correlations between gene expression levels across different samples, or experimental conditions. Usually, Pearson Correlation Coefficient (PCC) between all pairs of genes is computed based on their vectors of expression levels across different samples, or experimental conditions. Then, using a statistical method, a significant value of PCC (threshold) is determined. Two genes are linked if their PCC is higher than the threshold value. The choice of the threshold greatly influences the resulting topology of the co-expression network. Therefore, it is essential to

properly determine the appropriate threshold. For that purpose, a couple of methods have been proposed. Some methods use *prior* biological knowledge (e.g., known functional associations)[71] or network topology [72] to constrain the network construction; others use statistical comparison with randomised expression data [73], or random matrix theory [74]. However, these studies do not account for the number of samples (experimental conditions), which has been shown to greatly influence the choice of the correlation threshold. Hence, methods to overcome such limitations have been proposed. They rely on a combination of partial correlation and information theory approaches to determine a local correlation threshold for each pair of genes [75]. Unlike the single correlation threshold applied across the entire network, this approach allows for identification of more meaningful gene to gene associations [75].

- *A drug chemical similarity (DCS) network* represents similarities and differences between drugs' chemical structures. Each drug is composed of chemical sub-structures, which define its chemical fingerprint. These chemical fingerprints are usually represented by binary vectors whose coordinates encode the presence ("1") or absence ("0") of a particular sub-structure from a set of all known sub-structures. The chemical similarity between two drugs is computed based on the similarity between these vectors. Various measures for computing the similarity have been proposed [76, 77]. A commonly used one is Tanimoto coefficient [76]. The procedure for network construction is similar to the one of co-expression networks. First, the statistically significant similarity threshold is determined and then based on the threshold, links between drugs are constructed. Other types of drug-drug similarity networks have also been used in the network data integration literature. A frequently considered one is the *drug side-effect similarity (DSES) network*. Namely, clinical side-effects of each drug have been collected and stored in numerous databases. The side-effects provide a drug with a profile and allow for a construction of pairwise side-effect similarities between two drugs using the Jaccard Index [78].

- *Protein similarity networks* represent networks of proteins with similar sequences (PSeqS) [79] or structures (PStrS) [80]. Similarities between protein sequences are usually computed by using BLAST algorithm [81], whereas the similarities between three-dimensional protein structures are usually computed by using DaliLite algorithm [82].

Each of the above described biological network represents an important, component of the system at study. The networks often complement each other. For example, by comparing the number of common links between biological networks, many studies have reported a large overlap of the links between PPI and gene Co-Ex

networks [83], while a small overlap of links has been observed between PPI and GI networks [84]. Hence, in the tasks such as, protein function prediction, or disease gene prioritization, GI network has been shown to be a valuable complement to the PPI network, as it captures some hidden functional associations between genes that cannot be captured by experimental techniques designed for PPI identification [26, 70, 85].

2.3 Review of machine learning methods for biological data integration

Different data integration methods utilize different strategies to build integrative models. Different strategy often yields models with different levels of accuracy. Thus, choosing the right integration strategy for given data types is of great importance in many biological problems. According to many data integration studies, based on the stage at which the integration takes place, we can distinguish between the following three integration strategies [22, 86–91]:

- *Early* data integration strategy, combines the data sets into a single data set on which the data model is built. This often requires a transformation of the data sets into a common representation, which in some cases may result in information loss [22, 88].
- *Late* data integration strategy builds a model for each data set separately, then it combines these models into a unified model. Building models from each data set in isolation from others disregards their mutual relations, often resulting in a reduced performance of the final model [86, 88].
- *Intermediate* data integration combines data through inference of a joint model. This strategy has often been preferred due to its superior predictive accuracy reported in many studies (regardless of the chosen methods)[22, 86–89, 91]. This strategy does not require any data transformation and thus, it does not result in information loss.

Here, we review three widely used classes of machine learning techniques in the context of different biological problems, namely *Bayesian*, *Kernel-based (KB)*, and *Matrix factorization (MF) - based* techniques. First, we provide a general and brief introduction into these techniques, then we demonstrate their utility in data integration through some examples in systems biology and biomedicine. The special focus will be on the MF-based techniques, as this Thesis provides novel methods that are based on this type of technique.

2.3.1 Bayesian data integration

Bayesian methods build statistical models, in which starting from some *prior* knowledge, $p(\theta)$, about the model parameters, θ, they learn from the data, x, and compute the updated *posterior* probability, $p(\theta|x)$ making use of the *Bayes' rule*: $p(\theta|x) \propto p(\theta)p(x|\theta)$ [92]. Bayesian networks (BN) are one of the most commonly used methods for data integration, especially in problems such as *network inference* [93] and *gene function prediction* [25]. A BN is a directed acyclic graph, where nodes represent random variables (e.g., gene expression levels) and directed edges between the nodes represent conditional probabilities between pairs of variables. BNs provide an elegant way to represent the structure of the data and their sparsity enables a compact representation and computation of the *joint probability distribution* over the whole set of random variables.

Recently, BNs have been used as a suitable framework for integration and modelling of various types of biological data. Disparate data sources can be incorporated in either one of two steps of BN construction, i.e., *parameter learning*, or *structure learning*. Such networks play an important role in describing and predicting complex behaviour of a system supported by evidence from a variety of different biological data [94].

For example, Zhu et al. [95] combined gene expression data, expression of quantitative trait loci (eQTL), transcription factor binding site (TFBS) and PPI data to construct a causal, probabilistic network of yeast. In particular, they used the eQTL data to constrain the BN construction, so that *cis*-eQTL acting genes are considered to be parents of *trans*-eQTL acting genes. They tested the performance of the constructed BN in predicting GO categories and they demonstrated that the predictive power of the integrated BN is significantly higher than that of the BN constructed solely from the gene expression data [95]. A similar procedure has also been applied by Zhang et al. [96], who constructed a gene-regulatory network by integrating data from brain tissues of late-onset Alzheimer's disease patients.

One of the first studies that integrates clinical and patient-specific omics data is presented by Gevaert et al. [86]. They integrated the gene expression data of tissues from breast cancer patients whose clinical outcome was known. They constructed the BN with genes and *outcome* variable (representing clinical data) as nodes and used it for a classification task: they classified patients into good and poor prognosis groups. They compared the performance of BN in reproducing the known outcomes in three different strategies, early, late and intermediate (see above). They show that the intermediate strategy is the most accurate one. A similar conclusion was drawn by van Vliet et al. [87].

Most studies have utilised the simplest BN, the so called *naïve BN*, for combining multiple heterogeneous biological data and constructing an integrated gene-gene association network (also called a *functional linkage network*) [18, 19, 97, 98]. The structure of a naïve BN consists of a *class node* as a parent to all other independent nodes representing different data sources. Such a simple BN structure enables a much faster learning and inference. For example, in gene-gene association prediction, the class node may represent a set of interacting or non-interacting proteins, while the other variables in the naïve BN represent input biological data, often in pairwise format. In addition to the input data, a *gold standard* data (e.g., gene pairs with known functional relations, usually from GO) is used for learning, i.e., for constructing the probability distributions. The basic assumption of a naïve BN is that different data sources are conditionally independent, i.e., that the information in the data sets are independent given that a gene pair is either functionally associated or not.

Although simple, the naïve BNs have yielded good results in many data integration studies. For example, Troyanskaya et al. [25] developed a framework called MAGIC (Multi-source Association of Genes by Integration of Clusters) for gene function prediction. They integrated systems-level PPI and GI data along with GExD and TFBS data of *Saccharomyces cerevisiae*. By using GO as the gold standard, they demonstrated an increased accuracy of their method applied on all data sets, compared to its performance on each input data set separately. Furthermore, naïve BNs have demonstrated usability in patient-specific data integration. Namely, a recent study used a naïve BN to integrate gene expression and copy number variation (CNV) data of prostate and breast cancer patients [99]. Unlike other methods, this method successfully detected a new subtype of prostate cancer patients with extremely poor survival outcome. Moreover, unlike many data integration studies that force incompatible data types to be fused, this is the first study that systematically considered compatibility of input data sources. That is, the method was able to distinguish between concordant and discordant signals within each patient sample.

Bayesian networks are a good framework for biological network integration, because of their ability to capture noisy conditional dependence, that naturally exist in most of the biological data. However, they also have several disadvantages: 1) in the network inference problems, their sparse representation captures only important associations, while other associations are discarded; 2) their acyclic representation cannot be used for modelling networks with loops, which are important in many biological networks, as they represent control mechanisms; 3) the most important limitation of BNs is computational: learning and inference processes of BNs are computationally intractable on large data, which is a major reason why many studies

focus only on a small subset of nodes (genes) when constructing a BN.

2.3.2 Kernel-based data integration

Kernel-based (KB) methods belong to the class of *statistical machine learning methods* used for learning tasks, such as clustering, classification, regression, correlation, feature selection, etc. They work by mapping the input data, that is originally non-linear, into a higher dimensional space, called the *feature space*, where the data representation is hopefully linear and in which the pattern analysis is performed (e.g., for a data point, x_i, the mapping is given by a function ϕ: $x_i \rightarrow \phi(x_i)$). Such a mapping, is represented by a *kernel matrix* [91], $\mathbf{K}$; a symmetric, positive semi-definite matrix (i.e., having non-negative eigenvalues [100]) with entries $\mathbf{K}_{i,j} = k(x_i, x_j)$ representing similarities between all pairs of data points, x_i, x_j. Function $k(x_i, x_j)$ is called a *kernel function* and its explicit definition is the only requirement of the kernel method, while the mapping function ϕ and the properties of the feature space do not need to be explicitly specified. For example, given a set of proteins and their amino acid sequences, entries in the kernel matrix are usually generated by using the BLAST pairwise sequence alignment algorithm [81]. Hence, computing the protein embedding function, ϕ, and constructing the feature space, is not necessary. Other kernel functions for measuring similarities between two proteins include the *spectrum kernel* [101, 102], the *motif kernel* [103] and the *Pfam kernel* [104]. In addition to string data, molecular network data are frequently used in KB integration studies. Molecular networks are usually represented by *diffusion kernels*, which encode similarities between network's nodes [105], defined as: $\mathbf{K} = e^{-\beta \mathbf{L}}$, where $\beta > 0$ is the parameter that quantifies the degree of the diffusion and $\mathbf{L}$ is the network Laplacian (see Sec.2.1 for further details). The elements of the diffusion kernel matrix, $\mathbf{K}_{ij}$, quantify the closeness between nodes i and j in the network.

Methods using kernel-based approach include: support vector machines (SVM) [106], principal component analysis (PCA) [107], canonical correlation analysis (CCA) [108]. SVM classifiers have frequently been used for classification tasks in computational biology due to their high accuracy [109].

Kernel matrices represent a principled framework for representing different types of data including strings, sets, vectors, time series and graphs [110]. This property provides the KB methods with an advantage over the BN-based methods, as the BN-based methods are mostly suited for pairwise data sets due to easier construction of conditional probabilities in that case. However, choosing the "right" kernel function is not straightforward. Therefore, instead of constructing a single kernel, usually multiple candidate kernels for a single data set are constructed using

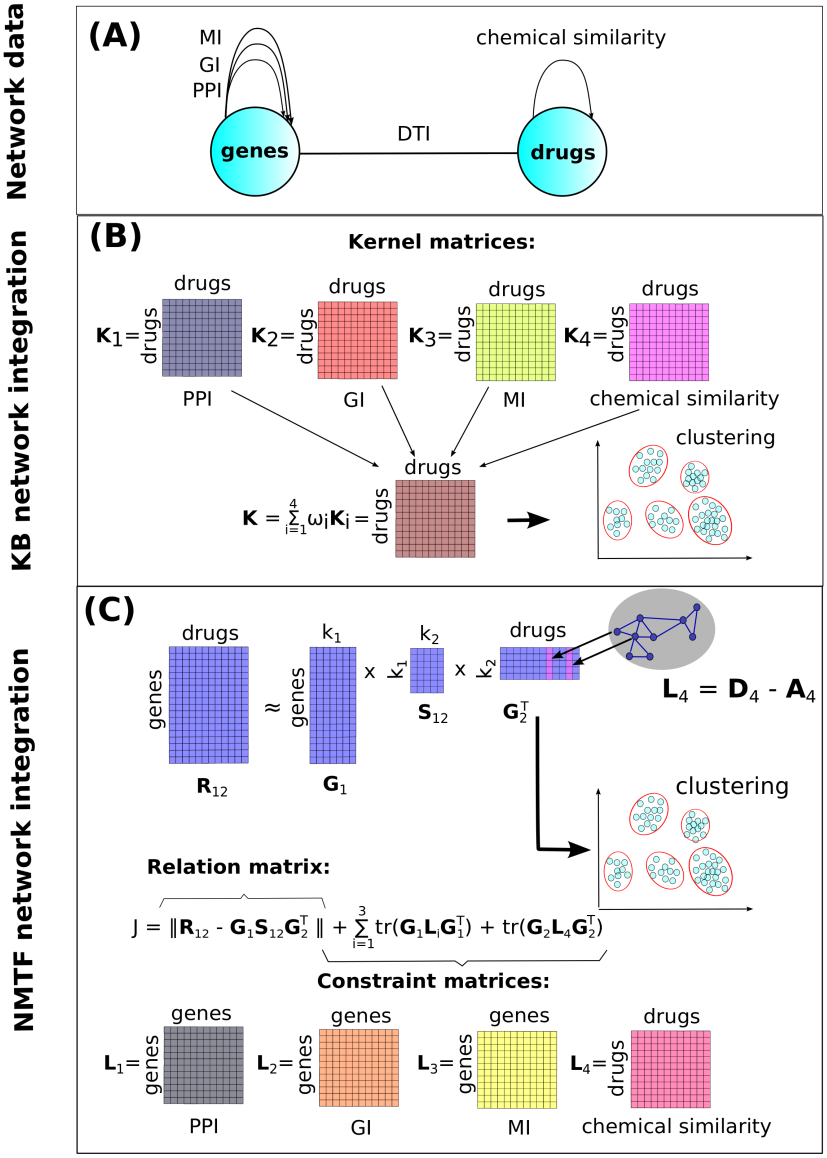


Figure 2.2: (A) Heterogeneous drug-target interaction network. *Inter-type* relations are represented by DTI network, whereas *intra-type* connections are represented by four molecular networks: PPI, GI and MI, and DCS network of drugs (see Sec.2.2 and Table 2.1). (B) An illustration of a KB data integration method for drug clustering. All kernel matrices are expressed in the drug similarity feature space based on the closeness between their targets (proteins) in each molecular network ($\mathbf{K}_1, \mathbf{K}_2$ and $\mathbf{K}_3$) and based on the similarity between their chemical structures ($\mathbf{K}_4$). All kernel matrices are linearly combined into a resulting kernel matrix, $\mathbf{K}$, on which the drug clustering is performed by using KB clustering methods. (C) An illustration of a NMTF-based data integration method for drug clustering: factorization of the DTI relation matrix under the guidance of molecular ($\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$) and chemical similarity ($\mathbf{L}_4$) constraint matrices. Drugs are assigned to clusters based on the entries in obtained $\mathbf{G}_2$ cluster indicator matrix. The figure is taken from Gligorijević and Pržulj [10] and modified.

different measures of similarity [111]. These kernels are then linearly combined into

one kernel that is then used for further analysis. Such an approach is called a *multiple kernel learning (MKL)* and has been shown to lead to a better performance than single KB methods, especially in genomic data analysis [112]. The mathematical basis for this approach is in the closure property of kernel matrices [110]. Namely, all elementary mathematical operations (e.g., addition, subtraction, multiplication) between two kernel matrices do not change the property of the final matrix (i.e., the positive semi-definite property stays preserved). This property provides the foundation for using KB methods for data integration by MKL. Namely, given a set of kernel matrices, $\{\mathbf{K}_1, \ldots, \mathbf{K}_n\}$, representing n different data sets, the single kernel matrix representing integrated data is obtained by a linear combination, $\mathbf{K} = \sum_{i=1}^{n} \omega_i \mathbf{K}_i$, where non-negative coefficients, ω_i, determine the weights of each data set and they are obtained through an optimisation procedure of the KB method [23, 113]. Namely, the non-negativity constraint imposed on weights reduces the optimisation problem to qudratically constrained quadratic problem (QCQP) [22]. The solution of this optimisation problem gives the weightning coefficients. Thus, the obtained weights, ω_i, allow us to distinguish between more and less informative data sets. Notice that all kernel matrices, $\mathbf{K}_i$, representing different data sets, need to be constructed over the same feature space to be correctly combined. In the case of heterogeneous data integration, this often requires data to be transformed, or projected onto the same feature space, which often results in information loss. This is the biggest drawback of KB data integration methods.

KB methods have first been proposed as a technique for data integration by Lanckriet et al. [22]. In that paper, they trained a *1-norm soft margin* SVM [2] to classify proteins into membrane or ribosomal groups. They constructed 7 different kernels representing three different types of data: amino acid sequences, gene expression and PPI data. They showed that the performance of their classifier is the highest when all data sets are taken into account. A similar approach was applied for function prediction of Baker's yeast proteins [23] and demonstrated that an SVM classifier trained on all data performs better than a classifier trained on any single type of data.

KB methods have also demonstrated their power in integrating molecular, structural and phenotypic data for drug repurposing. A recent study integrates three different layers of information represented in a drug-centered feature space [30]. Their kernel matrices represent: 1) drug chemical similarities based on their structures;

[2] *Soft margin* refers to the SVM method that allows for data points to be mislabeled. It is used when the hyperplane cannot cleanly separate data points onto -1 and $+1$ class. In that case, the soft margin method will choose a hyperplane that separates data points as cleanly as possible by introducing non-negative slack variables which measure the degree of misclassification. *1-norm* refers to the norm of slack variables introduced into the SVM objective function as penalisation terms [109].

2) drug similarities based on the positions of their targets in the PPI network; 3) and drug similarities based on the correlations between gene profiles under the drug influence. Based on the combination of these three kernel matrices and the existing drug classification, the authors trained the classifier and proposed the top misclassified drugs as new candidates for repurposing [30]. In a similar study, method called PreDR (**P**redict **D**rug **R**epurposing) for predicting novel drug-disease associations was proposed. A SVM classier was trained on integrated kernel matrices of drugs based on their chemical structure, target proteins and side-effect profiles [114].

Along with BN methods, KB methods can also be used for integration of clinical data with genomic data. An example study that integrates gene expression and clinical data of breast cancer patients is demonstrated by Daemen et al. [115]. They applied a Least Square SVM [116] on the combination of two patient-centric kernel matrices: the first is constructed based on patients' gene expression similarities, while the second is constructed based on similarities between patients' clinical variables. The authors reported 70% accuracy of their approach to correctly classify patients according to the appearance of distant sub-clinical metastases based on the primary tumor. They also demonstrated that both data sets, the expression and the clinical data, contribute to the performance of the classification.

From these examples, we can see that the KB data integration methods have several advantages. The main advantage is that they can integrate a wide range of data types. Moreover, a liner combination of kernels provides a selective way of accounting for data sets, by assigning lower weights to less informative and noisier data sets. However, a big disadvantage is that heterogeneous data sets need to be transformed into a common feature space to be properly integrated, which can lead to information loss (see example shown in Fig. 2.2 (A,B)). In the case of heterogeneous networks, the data transformation prevents modelling of multi-relational data, i.e., a simultaneous modelling of different types of relations in the data.

2.3.3 Matrix factorization-based data integration

In the last decade, matrix factorization (MF) methods have gained in popularity due to their efficiency in data clustering and missing data imputation. Specifically, the special focus has been on Non-negative Matrix Factorization (NMF) methods that have recently been proposed for data integration [88].

NMF method has been initially proposed for *dimensionality reduction* and *clustering* problems. It aims to find two low-dimensional, non-negative, latent feature matrices, $\mathbf{U} \in \mathbb{R}_+^{d \times k}$ and $\mathbf{V} \in \mathbb{R}_+^{n \times k}$, whose product provides a good approximation of the input non-negative data matrix, $\mathbf{X} \in \mathbb{R}_+^{d \times n}$, i.e., $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$. The method

was originally introduced by Lee and Seung [117] for parts-based decomposition of images, and later on it has found applications in many areas including document clustering, signal processing, bioinformatics, recommendation systems and social sciences. The non-negativity constraints provide matrix factors, $\mathbf{U}$ and $\mathbf{V}$, with a more meaningful interpretation than previous approaches, such as the principal component analysis (PCA) [107] and vector quantization (VQ) [118], while the choice of parameter $k \ll \min\{d, n\}$ (also called *the rank parameter*) provides a dimensionality reduction [119].

It has been shown that NMF is closely related to *K-means clustering*, one of the most widely used unsupervised learning algorithms [120]. Namely, NMF can be applied to clustering as follows: a set of n data points in d-dimensional space, represented by a data matrix $\mathbf{X}$, can be assigned to k clusters, based on the entries in $\mathbf{V} \in \mathbb{R}_+^{n \times k}$. In particular, each data point, i, is assigned to cluster, j, if $\mathbf{V}_{ij}$ is the largest entry in row i of matrix $\mathbf{V}$. This procedure is called *hard clustering*, as each data point belongs to exactly one cluster [121]. The non-negativity of $\mathbf{V}$ matrix is exactly what provides the clustering interpretation, and thus the $\mathbf{V}$ matrix is also called a *cluster indicator* matrix.

The biggest application of NMF comes with its extension to handle heterogeneous data. Namely, the above described NMF can only be used for homogeneous data clustering (clustering object of the same data type). Therefore, the formalism was further extend by Ding et al. [122] to simultaneously clustering rows and columns (i.e., bi-cluster) of the data matrix by defining Non-negative Matrix Tri-Factorization (NMTF). For example, given a data matrix, $\mathbf{R}_{12} \in \mathbb{R}_+^{n_1 \times n_2}$, encoding relations between two sets of objects of different types (e.g., adjacency matrix of DTI bipartite network representing interactions between n_1 genes and n_2 drugs, see example in Fig. 2.2 (A)), NMTF, decomposes matrix $\mathbf{R}_{12}$ into a product of three non-negative matrix factors as follows: $\mathbf{R}_{12} \approx \mathbf{G}_1 \mathbf{S}_{12} \mathbf{G}_2^T$, where $\mathbf{G}_1 \in \mathbb{R}_+^{n_1 \times k_1}, \mathbf{G}_2 \in \mathbb{R}_+^{n_2 \times k_2}$ are, in analogy with matrix $\mathbf{V}$ in NMF, the cluster indicator matrices of the first and the second data set, respectively, and $\mathbf{S}_{12} \in \mathbb{R}_+^{k_1 \times k_2}$ is a low-dimensional representation of the initial matrix. In analogy with NMF method, rank parameters k_1 and k_2 correspond to numbers of clusters in the first and the second data set (i.e., genes and drugs respectively). In addition to co-clustering, NMTF can also used for *matrix completion* [123]. Namely, after obtaining low-dimensional matrix factors, the *reconstructed data matrix*, $\hat{\mathbf{R}}_{12} = \mathbf{G}_1 \mathbf{S}_{12} \mathbf{G}_2^T$, is more complete then the initial data matrix, $\mathbf{R}_{12}$, featuring new entries, unobserved in the data, that emerged from the low-dimensional latent structure captured by the non-negative matrices. Therefore, NMTF provides a unique approach for modelling multi-relational heterogeneous network data and predicting new, previously unobserved links.

The problem of finding optimal low-dimensional non-negative, latent feature

matrices whose product is equal to the initial data matrix is known to be NP-hard [124]. Thus, heuristic algorithms for finding approximate solutions have been proposed [125]. They involve solving an *optimisation problem* that minimises the distance between the input data matrix and the product of low-dimensional matrix factors. The most common measure of the distance used in construction of the objective (cost) function is the *Frobenius norm* (also called the *Euclidean norm*) [100]. Hence, the objective function to be minimised can be defined as follows [3]: $\min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} J = \min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} \parallel \mathbf{R}_{12} - \mathbf{G}_1 \mathbf{S}_{12} \mathbf{G}_2^T \parallel_F^2$. The low-dimensional matrix factors, $\mathbf{G}_1, \mathbf{G}_2$ and $\mathbf{S}_{12}$, are computed by using *iterative update rules* derived by applying standard procedures from *constrained optimisation theory* [127]. These update rules ensure decreasing behaviour of the objective function, J, over iterations. The most popular rules are *multiplicative update rules*, which preserve the non-negative property of the matrix factors through update iterations. They start with randomly initialised matrix factors and iteratively update them until the convergence criterion is met [122, 128]. Other ways and strategies for matrix initialization that lead to much faster convergence have been proposed [125, 129].

The NMF optimization problems belong to the group of *non-convex* optimisation problems (i.e., the objective function, J, is a non-convex function of its variables) [125]. Unlike *convex* optimisation problems, which are characterised by the global minimum solution and whose algorithms scale well with the problem size [130], *non-convex* optimisation problems face a range of difficulties, including finding the global minimum (and thus the unique solution) and a very slow convergence to a local minimum. Nevertheless, even a local minimum solution of NMF has been shown to have meaningful properties in many data mining applications [125].

Depending on the type of constraints imposed on the low-dimensional latent feature matrices, various variants and improvements of the NMTF method have been proposed in the literature [131]. One of the most widely used one is the *graph regularized NMTF (GNMTF)* [132] (see Fig. 2.2, C). Namely, the NMTF model is extended by incorporating additional, *prior* information about pairwise relations of data points into the objective function to guide the bi-clustering process. Namely, in many studies, the data points itself can lie on a low-dimensional manifold, or have pairwise structure present in an external data source often represented by a network. For example, in Fig. 2.2 (A), in addition to *inter-type* drug-gene relations represented by relation matrix, $\mathbf{R}_{12}$, both biological entities, genes and drugs, are characterised by *intra-type* relations represented by different networks, i.e., molecular networks connecting genes and a chemical similarity network connecting drugs, respectively.

[3] without any loss of generality, we can remove the non-negativity constraint imposed on $\mathbf{S}$ matrix. In this case, the method is referred as Semi-Nonnegative Matrix Tri-Factorization, or Semi-NMTF [126]

These relations are represented by Laplacian matrices, $\mathbf{L}_i$, and they are incorporated into the objective function as constraints to guide the bi-clustering of genes and drugs, by enforcing two connected genes (or two connected drugs) to belong to the same gene (or drug) cluster, respectively (Fig. 2.2, C). For instance, the last two terms in the formula displayed in Fig. 2.2 (C) represent graph-regularization terms through which these constraint matrices are incorporated into the objective function. Thus, graph-regularization offers an elegant way to integrate additional data sets, and analyse heterogeneous multi-relational networks (see Section 2.1 where this network type was first introduced).

MF-based techniques provide a principled way to integrate many different data sets. The basic principle behind all MF-based data integration methods is that they jointly factorize matrices representing different data sets by sharing one or more low-dimensional latent feature matrices across the decompositions. For example, one of the first methods of this kind was introduces by Zhang et al. [133], and it was proposed for simultaneous decomposition od 2D images, $\mathbf{X}_k \approx \mathbf{U}\mathbf{V}_k$, by sharing one of the latent matrices, i.e., $\mathbf{V}_k$. The same decomposition was later used in biology for for discovering microRNA-gene regulatory modules [134], for finding integrative modules by integrating genomic data [135], and for analysing multiple gene expression data sets [136]. On the other hand, simultaneous tri-factorization, for handling any number of heterogeneous multi-relational data sets (e.g., sets of interconnected genes, drugs, diseases, etc.), have been proposed by Wang et al. [137]. In their method, they simultaneously tri-factorize relation matrices, $\mathbf{R}_{ij} \approx \mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T$, representing pairwise relations between data points of different types, i and j. In addition to simultaneous decomposition of relation matrices, $\mathbf{R}_{ij}$, Wang et al. [138] further constrained construction of $\mathbf{G}_i$ matrices by introducing constraint matrices representing intrinsic pairwise relations of data points of each data set, i. Within the same objective function, they introduced graph-regularization terms with the goal to guide the co-clustering process of each data set. This method, with minor mathematical improvements, was proposed to be used for heterogeneous data integration (fusion) by Žitnik and Župan [88]. The key ingredients of their approach, termed *Data Fusion by Matrix Factorization (DFMF)*, are low-dimensional factors, $\mathbf{G}_i$, that are **shared** across the decomposition of all relation matrices, ensuring the influence of all data sets onto the resulting model. A pioneering application was demonstrated on predicting new disease-disease associations [70] by fusing large-scale heterogeneous networked data, including gene–DO-term, gene-drug and gene-GO-term *inter-type* associations (where data types are genes, DO terms, GO terms and drugs) together with *intra-type* associations represented by 5 different molecular interaction networks (PPI, GI, MI, Co-Ex and cellular signalling) connecting genes, a DSES network connecting drugs, and GO and DO semantic relations connecting GO and DO terms, respectively.

Hence, NMF methods provide a principled framework for integration of both *homogeneous* and *heterogeneous* data. Moreover, MF-based methods have numerous advantages over the other two methods outlined in this section. First, they do not require any data transformation, or any special matrix construction, but instead, they integrate networks naturally represented by adjacency matrices. This drastically reduces chances for information loss. Second, the great accuracy of the method, whose superiority over KB has been demonstrated, stems from the intermediate integration strategy [88]. Finally, the biggest advantage of the method is in its ability to simultaneously model all types of relations in the data, i.e., to simultaneously cluster and create predictive models of all types of data without any data transformation. In contrast, KB methods can only model only one type of data at a time by transforming all data sources into a common feature space. In Fig. 2.2, we illustrate the differences between these two methods. Unlike KB, NMTF can be used to bi-cluster genes and drugs simultaneously, as well as to create a model of gene-drug relations, by using evidence from all available networks. In contrast, KB methods can only be used to classify one entity at the time (either drugs or genes or their relations).

More detailed literature reviews about the MF-based methods for specific biological problems studied in this Thesis are given in the introduction parts of the corresponding chapters.

Chapter 3

Integration and analysis of heterogeneous multi-relational networks

In this chapter we focus on the tri-factorization method applied on *heterogeneous multi-relational* networks. The methodological part of this chapter is largely based on works by Wang et al. [137] and Wang et al. [138]. However, the method is further extended to address three seemingly different biological problems. The problems are presented in order according to the complexity of the underlying objective function used for modeling the data in these problems, from the least complex to the most complex objective function.

In the fist problem (presented in Section 3.1), we deal with gene function prediction in yeast model organism using heterogeneous network setup. Here, we integrate various types of molecular interaction networks along with existing gene-GO annotations and GO hierarchical associations to postulate for new annotations. We also ask how much of GO hierarchy can be reconstructed solely from topology of molecular networks, and specifically which network type contributes the most to the quality of our predictions. The results of this section have previously been published in Gligorijević et al. [26]. In the second problem (presented in Section 3.2), we shift our focus on precision medicine, and propose a general and versatile framework for integrating cancer genomic data along with molecular networks to address the three key questions, patient stratification, cancer driver prediction and drug repurposing. The results of this section have previously been published in Gligorijević et al. [139]. Finally, the third problem (presented in Section 3.3) deals with multiple alignment of PPI networks of 5 different species. Here, we integrate PPI network along with protein sequence similarity of different species to find structurally and functionally

consistent clusters of proteins across different species. The results of this section have previously been published in Gligorijević et al. [140].

3.1 Network integration for protein function prediction and GO reconstruction

Recently, a shift was made from using GO to evaluate molecular network data to using these data to construct and evaluate GO. Dutkowski et al. [85] provide the first evidence that a large part of GO can be reconstructed solely from topologies of molecular networks. Motivated by this work, we develop a novel data integration framework that integrates multiple types of molecular network data to reconstruct and update GO. We ask how much of GO can be recovered by integrating various molecular interaction data.

We introduce a computational framework for integration of various biological networks using Graph-regularized Non-negative Matrix Tri-Factorization (GNMTF). It takes all network data in a matrix form and performs simultaneous clustering of genes and GO terms, inducing new relations between genes and GO terms (annotations) and between GO terms themselves. To improve the accuracy of our predicted relations, we extend the integration methodology to include additional topological information represented as the similarity in wiring around non-interacting genes.

Surprisingly, by integrating topologies of baker's yeast protein-protein interaction, genetic interaction and co-expression networks, our method reports as related 96% of GO terms that are directly related in GO. The inclusion of the wiring similarity of non-interacting genes contributes 6% to this large GO-term association capture. Furthermore, we use our method to infer new relationships between GO terms solely from the topologies of these networks and validate 44% of our predictions in the literature. In addition, our integration method reproduces 48% of cellular component, 41% of molecular function and 41% of biological process GO terms, outperforming the previous method in the former two domains of GO. Finally, we predict new GO annotations of yeast genes and validate our predictions through genetic interactions profiling.

3.1.1 Introduction

In many areas of biomedical research ontologies play an important role in unification of knowledge as a hierarchy of terms and their mutual relationships. Among widely

used ontologies is Gene Ontology (GO), which describes genes and gene products in terms of their associated *biological process (BP)*, *molecular function (MF)* and *cellular component (CC)* [21]. GO is a current major source of information for annotating genes and proteins across various species and providing tools for systematic assessment of experimental gene sets via enrichment analysis.

Since its foundation, GO has been growing in size and complexity containing today vast amounts of annotated biological data. Initially, GO was manually curated by domain experts and members of the research and annotation communities. However, due to their inconsistency in translation to GO terms and relations, manual curations have encountered many difficulties [141]. Additionally, rapid development of technologies for biological data acquisition has resulted in an accumulation of biological data exceeding our ability to interpret [142].

To overcome these problems, many computational tools for automatic gene and protein annotation have been devised. Much effort has been invested in assessing the accuracy of such annotation predictions [143]. Methods for gene annotation prediction have either followed approaches that transfer annotations from well-observed to partially observed genes based solely on sequence similarity [144] or approaches that directly predict function of unknown genes using machine learning methods [145]. Recent methodologies focus more on integration of distinct biological data sources which contribute to more accurate predictions of gene annotation.

The availability of genomic-level information from high-throughput measurements of genetic and protein interactions, mRNA expression profiles and metabolic pathways has created new opportunities for function prediction. A major challenge is how to integrate all these diverse data to predict annotations of yet unannotated proteins. Among the widely used computational methods addressing this problem are *Bayesian reasoning* [142], *network-based analysis* [146, 147], *kernel-based statistical learning* [22] and *matrix factorization-based data fusion*[148]. All these methods have demonstrated that the integration of complementary biological data significantly improves accuracy of gene function annotation prediction.

Recent work incorporated large gene and protein interaction networks into a probabilistic clustering procedure to reconstruct the Gene Ontology [85]. It identified new terms and relations that were missing from GO based solely on network topology. This work provides evidence that a large part of GO can be reconstructed using only topologies of molecular networks.

In this work we propose a new data integration method for prediction of GO term annotations of unannotated genes and finding new relations between existing GO terms purely from network topology. The method is based on GNMTF for heterogeneous data clustering [137, 138]. It has been used for prediction of disease

associations [70], identification of cancer subtypes [149], predicting protein-protein interactions [28], and detecting phenotype-gene associations [150].

Here, we extend this method to take multiple types of molecular network data and utilize them to reconstruct and update GO with new information. We apply our method to *Saccharomyces cerevisiae* data used by [85]: protein-protein interaction (PPI) network, genetic interaction (GI) network, gene co-expression (Co-Ex) network and integrated functional network known as YeastNet [151].

Our method takes all data in a matrix form and performs simultaneous clustering of genes and GO terms inducing new associations between genes and GO terms and between GO terms themselves. We extend the integration methodology to include similarity in wiring around non-interacting genes. We measure this by distance graphlet degree vectors (GDV) [152]. Graphlets and graphlet-based measures have bridged molecular network topology and biological function. For instance, simple homogeneous clustering of proteins in a PPI network based on the GDV similarity has revealed groups of proteins with a common biological function [153, 154]. Therefore, we add these to incorporate more topology into the integration process and improve accuracy of predictions. Using various measures for assessing the quality of our prediction, we systematically examined the contribution of these additional topological constraints to GO prediction. Graphlet-based similarity has not been exploited in any of the previous network integration approaches.

Surprisingly, we find that our method can successfully reconstruct almost the entire GO by using solely topology of molecular interaction networks. Furthermore, we predict new GO term associations and gene annotations from integrated topologies of molecular interaction network and validate our predictions.

3.1.2 Methods

We use GNMTF (see Section 2.3 for a brief introduction to this methods) to integrate all available information on genes and GO terms presented in the network form, including both *inter-type relations* (gene-GO term relations) and *intra-type constraints* (molecular interactions between genes and hierarchical relations between GO terms) [137]. We factorize gene-GO term annotation matrix by incorporating GO graph and different types of molecular networks as constraints into the GNMTF objective function. From the obtained low-dimensional latent feature matrices, we infer new relations between genes and GO terms (i.e., new gene annotations) that were not present in the initial data, and we also infer new relations between GO terms.

The relations between n_1 genes and n_2 GO terms (i.e., also known as *inter-type*

relations) are represented by $\mathbf{R}_{12} \in \mathbb{R}^{n_1 \times n_2}$ matrix, with $\mathbf{R}_{12}(i,j) = 1$ if a gene i is annotated by a GO term j, and $\mathbf{R}_{12} = 0$ otherwise. Simultaneous clustering of genes and GO temrs can be seen as a solution of the semi-nonnegative matrix tri-factorization (NMTF) [1] problem where the relation matrix, $\mathbf{R}_{12} \in \mathbb{R}^{n_1 \times n_2}$, is approximated as the product of three low-dimensional latent feature matrices:

$$\mathbf{R}_{12} \approx \mathbf{G}_1 \mathbf{S}_{12} \mathbf{G}_2^T, \tag{3.1}$$

where non-negative $\mathbf{G}_1 \in \mathbb{R}_+^{n_1 \times k_1}$ and $\mathbf{G}_2 \in \mathbb{R}_+^{n_2 \times k_2}$ correspond to *soft clustering* membership matrices of genes and GO terms, respectively, and $\mathbf{S}_{12} \in \mathbb{R}^{k_1 \times k_2}$ corresponds to compressed low-dimensional version of the initial relation matrix. Rank parameters, k_1 and k_2, are chosen to be much smaller than the corresponding matrix dimensions ($k_1 \ll n_1$, $k_2 \ll n_2$), and they also represent the number of clusters of genes and GO terms, respectively.

To compute low-dimensional matrices, NMTF algorithm minimizes the following objective function:

$$\min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} J = \min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} \parallel \mathbf{R}_{12} - \mathbf{G}_1 \mathbf{S}_{12} \mathbf{G}_2^T \parallel^2 \tag{3.2}$$

This objective function can be further used to incorporate network constraints (also known as *intra-type* constraints) whose violation causes penalties. Namely, biological entities, such as genes and proteins engage in various molecular interactions, whereas, GO terms are connected through GO hierarchical relationships. We represent these as networks and incorporate them into the objective function (Eq. 3.2) as network Laplacians [150, 155]:

$$\min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} J = \min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} \Big(\parallel \mathbf{R}_{12} - \mathbf{G}_1 \mathbf{S}_{12} \mathbf{G}_2^T \parallel^2 + tr(\mathbf{G}_1^T \mathbf{L}_1 \mathbf{G}_1) + tr(\mathbf{G}_2^T \mathbf{L}_2 \mathbf{G}_2) \Big) \tag{3.3}$$

where, $\mathbf{L}_1 = \mathbf{D}_1 - \mathbf{A}_1$ and $\mathbf{L}_2 = \mathbf{D}_2 - \mathbf{A}_2$ are Laplacian matrices of a gene interaction network and GO graph, respectively, and tr denotes the trace of a matrix. The first Laplacian-based (penalty) term encourages the connected genes (proteins) in the network to be placed into the same cluster and similarly the second term encourages the hierarchically related GO terms to be placed into the same cluster.

This optimization problem (Eq. 3.3) is known as *GNMTF* problem. Its solution produces two low-dimensional latent feature matrices, $\mathbf{G}_1$ and $\mathbf{G}_2$, that can be interpreted as *soft cluster indicator matrices* of genes and GO terms respectively.

[1] "semi-nonnegative" refers to the fact that $\mathbf{S}_{12}$ matrix is not required to be nonnegative

Specifically, matrix $\mathbf{G}_1$ is used to assign n_1 genes to k_1 clusters, and similarly, matrix $\mathbf{G}_2$ is used to assign n_2 GO terms to k_2 clusters. Here we adopt the *consensus clustering* procedure proposed by Brunet et al. [156]. Namely, we first use the clustering assignment procedure, described in Section 2.3, to assign genes and GO terms to clusters of genes and GO terms, respectively. For example, gene i is placed into a cluster j if $\mathbf{G}_1(i,j)$ is the largest entry in row i. Then, we construct binary *connectivity matrix* of genes, $\mathbf{C}_1$, of size $n_1 \times n_1$ with entry $\mathbf{C}_1(p,q) = 1$ if genes p and q belong to the same cluster and $\mathbf{C}_1(p,q) = 0$ otherwise. The procedure can be repeated (usually with different initialization strategy, see Section 3.1.5 for more details) resulting in an *consensus connectivity matrix*, $\overline{\mathbf{C}}_1$, from which the final clusters can be extracted. Similar procedure applies to GO terms.

3.1.3 Integration of various molecular networks

Most of the biological data sets include various types of interactions (i.e., constraints) over the same set of entities. For instance, genes might interact via genetic interactions and they also might be related based on the correlation of their expression profiles. The former is known as a GI network, and the latter as a Co-Ex network (see Table 2.1 for abbreviations). To properly integrate this information into the clustering procedure, we make an improvement to the GNMTF optimization problem. We extend it to take into account multiple types of connections between genes. Suppose we have a set of N adjacency matrices: $\{\mathbf{A}_1^{(1)}, \mathbf{A}_1^{(2)}, \ldots, \mathbf{A}_1^{(N)}\}$, representing N data network types (modalities) of gene relations. We add this matrices as additional Laplacian-based constraints, $\mathbf{L}_1^{(\alpha)} = \mathbf{D}_1^{(\alpha)} - \mathbf{A}_1^{(\alpha)}$, into our objective function (Eq. 3.3), and obtain the following:

$$\min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} J = \min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} \Big(\parallel \mathbf{R}_{12} - \mathbf{G}_1 \mathbf{S}_{12} \mathbf{G}_2^T \parallel^2 + \sum_{\alpha=1}^{N} tr(\mathbf{G}_1^T \mathbf{L}_1^{(\alpha)} \mathbf{G}_1) + tr(\mathbf{G}_2^T \mathbf{L}_2 \mathbf{G}_2) \Big) \tag{3.4}$$

Integration of all available gene (protein) interaction networks is believed to lead to better and more accurate predictions of new gene annotations. For example, the integration of all available human molecular networks has already shown to yield a successful reproduction of the existing and prediction of new associations between diseases [70].

Unlike previous works where only directly connected genes are considered as pairwise constraints [28, 70, 134, 149, 150], our approach takes a step further by incorporating additional pairwise constraints in the form of topological similarity

between genes in a network that are not necessarily linked. Here, we use the topological similarity measure based on *graphlet degree vectors (GDVs)*. Graphlets are small non-isomorphic induced substructures of a large network (Fig. B.1 panel A in Appendix B) [44]. There are 30 graphlets containing 2-5 nodes. By taking into account the symmetries between nodes in a graphlet, we can distinguish between 73 *automorphic orbits*. Counting how many times a particular node touches any of 73 different orbits, we can define a 73-dimensional graphlet degree vector (GDV) (see Figure B.1 in Appendix B). GDV vector represents local structural properties of a node and therefore, it can be used to compare topologies around nodes in a network (Fig. B.1 panel B). For that purpose a measure of distance between nodes u and v is introduced as [153]:

$$D(u,v) = \frac{\sum_{i=1}^{73} D_i(u,v)}{\sum_{i=1}^{73} w_i}, \tag{3.5}$$

where $D_i(u,v)$ is defined as a logarithmic distance between nodes' i-th orbits:

$$D_i(u,v) = w_i \times \frac{|\log(u_i+1) - \log(v_i+1)|}{\log(max\{u_i,v_i\}+2)}$$

To take into account mutual dependencies between orbits, a weight $w_i = 1 - \frac{log(o_i)}{log(73)}$ is assigned to each orbit $i \in \{0,\ldots,72\}$. The weight, w_i, measures to which extent orbit i is affected by other orbits. Higher weights are assigned to orbits that are less affected by the orbits while lower weight are assigned to orbits that are affected by many other orbits. The number of orbits that affect orbit i is given by o_i. Using the distance measure defined in equation (3.5), *GDV similarity* between nodes u and v is measured as:

$$\sigma(u,v) = 1 - D(u,v) \tag{3.6}$$

GDV similarity measure has been used for predicting biological function of unclassified proteins [153], classification of cancer and non-cancer genes [154] and prediction of new cardiovascular disease genes [157].

Here, we include GDV similarity measure into our objective function (Equation 3.4) as follows. For each of the given biological network $\beta = \{1,\ldots,N\}$, we construct a GDV similarity matrix $\boldsymbol{\sigma}^{(\beta)} \in \mathbb{R}^{n\times n}$ according to equation 3.6. Then, by computing a statistically significant threshold for topological similarity of two nodes in each of the GDV similarity matrices, we consider only gene pairs with GDV similarity higher than the computed threshold (see Figure B.2 in Appendix B):

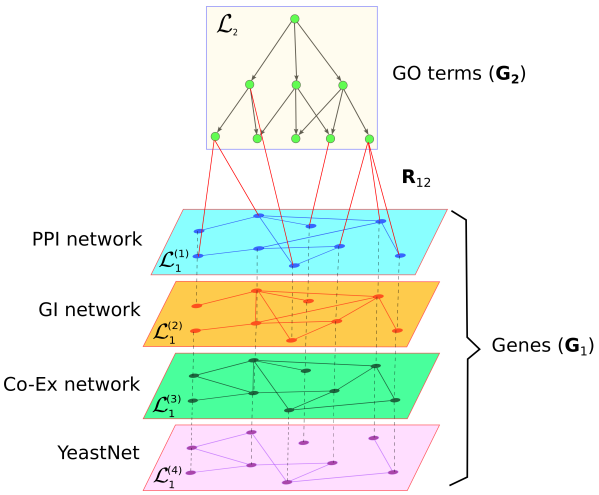


Figure 3.1: Schematic representation of multi-relational network used in this study. Two types of objects are represented: *genes* interconnected via four types of interaction networks (PPI, GI, Co-Ex and YeasNet) and *GO terms* interconnected via directed, semantic relations from GO hierarchy. The figure is taken from Gligorijević et al. [26].

$$\mathbf{\Sigma}^{(\beta)}(u,v) = \begin{cases} 1, & \text{if } \boldsymbol{\sigma}^{(\beta)}(u,v) \geq \boldsymbol{\sigma}^{(\beta)}_{threshold} \\ 0, & \text{if } \boldsymbol{\sigma}^{(\beta)}(u,v) < \boldsymbol{\sigma}^{(\beta)}_{threshold} \end{cases}$$

Topological similarity constraints are again implemented into the objective function (Equation 3.4) through Laplacian regularization:

$$\begin{aligned} \min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} J = \min_{\mathbf{G}_1 \geq 0, \mathbf{G}_2 \geq 0} \Big(& \parallel \mathbf{R}_{12} - \mathbf{G}_1 \mathbf{S}_{12} \mathbf{G}_2^T \parallel^2 + \\ & + \sum_{\alpha=1}^{N} tr(\mathbf{G}_1^T \mathbf{L}_1^{(\alpha)} \mathbf{G}_1) + \sum_{\beta=1}^{N} tr(\mathbf{G}_1^T \mathbf{\Lambda}_1^{(\beta)} \mathbf{G}_1) + tr(\mathbf{G}_2^T \mathbf{L}_2 \mathbf{G}_2) \Big) \end{aligned} \tag{3.7}$$

where, $\mathbf{\Lambda}^{(\beta)} = \mathbf{\Delta}^{(\beta)} - \mathbf{\Sigma}^{(\beta)}$ is a Laplacian of $\mathbf{\Sigma}^{(\beta)}$ matrix and $\mathbf{\Delta}^{(\beta)}$ is the diagonal matrix with entries equal to the row summation of $\mathbf{\Sigma}^{(\beta)}$ matrix.

3.1.4 Multiplicative update algorithm

We extend the original algorithm by Wang et al. [137] for optimizing objective function in equation 3.3 to handle the additional network regularization terms in equation 3.7. The modified multiplicative update rules for low-dimensional matrices $\mathbf{G}_1$, $\mathbf{G}_2$ and $\mathbf{S}_{12}$ [137] are as follows:

$$\mathbf{S}_{12} \leftarrow (\mathbf{G}_1^T\mathbf{G}_1)^{-1}\mathbf{G}_1^T\mathbf{R}_{12}\mathbf{G}_2(\mathbf{G}_2^T\mathbf{G}_2)^{-1} \tag{3.8}$$

$$\mathbf{G}_1(i,j) \leftarrow \mathbf{G}_1(i,j)\sqrt{\frac{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}_{12}^T)_{ij}^+ + [\mathbf{G}_1(\mathbf{S}_{12}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S}_{12})^-]_{ij} + [\sum_\beta(\mathbf{A}_1^{(\beta)} + \mathbf{\Sigma}_1^{(\beta)}))G_1]_{ij}}{(\mathbf{R}_{12}\mathbf{G}_2\mathbf{S}_{12}^T)_{ij}^- + [\mathbf{G}_1(\mathbf{S}_{12}^T\mathbf{G}_2^T\mathbf{G}_2\mathbf{S}_{12})^+]_{ij} + [\sum_\beta(\mathbf{D}_1^{(\beta)} + \mathbf{\Delta}_1^{(\beta)})\mathbf{G}_1]_{ij}}} \tag{3.9}$$

$$\mathbf{G}_2(i,j) \leftarrow \mathbf{G}_2(i,j)\sqrt{\frac{(\mathbf{R}_{12}\mathbf{G}_1\mathbf{S}_{12})_{ij}^+ + [\mathbf{G}_2(\mathbf{S}_{12}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}_{12}^T)^-]_{ij} + [\mathbf{A}_2\mathbf{G}_2]_{ij}}{(\mathbf{R_{12}}\mathbf{G_1}\mathbf{S_{12}})_{ij}^- + [\mathbf{G}_2(\mathbf{S}_{12}\mathbf{G}_1^T\mathbf{G}_1\mathbf{S}_{12}^T)^+]_{ij} + [\mathbf{D}_2\mathbf{G}_2]_{ij}}} \tag{3.10}$$

where we use + and − sign in superscripts to denote nonnegative matrices $\mathbf{M}^+$ and $\mathbf{M}^-$ of a matrix $\mathbf{M}$ defined as: $\mathbf{M}^+ = \frac{|\mathbf{M}|+\mathbf{M}}{2}$ and $\mathbf{M}^- = \frac{|\mathbf{M}|-\mathbf{M}}{2}$. The algorithm starts by randomly initializing matrices $\mathbf{G}_1$ and $\mathbf{G}_2$, which are iteratively updated to minimize objective function in equation (3.7). The rigorous proof of the correctness and convergence of these update rules can be found in [137]. Under these update rules, the objective function J (equation 3.7) is guaranteed not to increase. Hence, we look at the change in the objective function between two consecutive iterations and define the *stopping criterion* as: $\frac{|J_n - J_{n-1}|}{|J_{n-1}|} < \epsilon$. In all our runs, parameter ϵ is set to 10^{-5} which was shown to be significant to minimize the objective function. Compared to the probabilistic clustering approach for GO reconstruction presented by Dutkowski et al. [85], our approach is computationally more demanding due to slow convergence of multiplicative update rules. However, our approach is more general, since it can integrate any number and type of heterogeneous data, that could lead to more accurate predictions.

3.1.5 Predicting associations between GO terms

Each factorization run produces matrix factors: $\mathbf{G}_1$ related to gene set, and $\mathbf{G}_2$ related to GO terms. We use $\mathbf{G}_2$ factor to construct connectivity matrix $\mathbf{C}_2$ as described in the Section 3.1.2. Relations between GO terms are obtained from the connectivity matrix. To assess reliability and robustness of GO term associations prediction, we use the stochastic property of our algorithm. We perform multiple runs with the same rank parameters and different initial random initializations and construct a set of 20 different connectivity matrices: $\{\mathbf{C}_2^{(1)}, \ldots, \mathbf{C}_2^{(20)}\}$. Then, we compute the *consensus matrix*, $\overline{\mathbf{C}}_2$, defined as the average over all connectivity matrices. Thereby entries in the consensus matrix range from 0 to 1 and they can be interpreted as probabilities that two GO terms, GO_i and GO_j, belong to the

same cluster. To predict new GO term associations, we are only interested in values of probability equal to one, because they correspond to the case of hard clustering, in which there is no overlap between clusters and hence, there is no ambiguity in predicted GO term associations. The complete algorithm for prediction of new GO term association is summarized in Algorithm 1.

Algorithm 1 GO term associations prediction
Input: Annotation matrix: $\mathbf{R}_{12}$; Constraint matrices: $\mathbf{\Lambda}_1^{(\beta)}$, $\mathbf{L}_1^{(\beta)}$, for networks $\beta \in \{1,2,3,4\}$; $\mathbf{L}_2^{\beta}$ for Gene Ontology; Rank parameters: k_1 and k_2
Output: Consensus matrix $\overline{\mathbf{C}}_2$

for $i \in [1,20]$ **do**
  Initialize $\mathbf{G}_1$ and $\mathbf{G}_2$
  while not $|J_n - J_{n-1}|/|J_{n-1}| < \epsilon$ **do**
    Update $\mathbf{S_{12}}$ using equation (3.8) while keeping fixed $\mathbf{G}_1$ and $\mathbf{G}_2$
    Update $\mathbf{G}_1$ using equation (3.9) while keeping fixed $\mathbf{G}_2$ and $\mathbf{S_{12}}$
    Update $\mathbf{G}_2$ using equation (3.10) while keeping fixed $\mathbf{G}_1$ and $\mathbf{S_{12}}$
  Compute connectivity matrix $\mathbf{C}_2^{(i)}$ for GO terms using $\mathbf{G}_2$ for class assignment
Compute the average connectivity matrix as: $\overline{\mathbf{C}}_2 = \frac{1}{20}\mathbf{C}_2^{(i)}$
Extract new GO term relations:
$\mathcal{G} = \left\{(GO_i, GO_j) \mid \forall GO_i, \forall GO_j \in \{all\ GO\ terms\} \wedge \overline{\mathbf{C}}_2(i,j) = 1\right\}$

To assess the statistical significance of GO term associations, we compute the p-value in the following way. First, we remove any prior knowledge on GO term relations (i.e., we remove matrix $\mathbf{L}_2$). Then, we run our algorithm 100 times, each time with different relations matrix obtained by permuting the entries of the original relations matrix, $\mathbf{R}_{12}$. In total, we obtain $100 \times 20 = 2000$ different connectivity matrices. We define the p-value of a particular GO term association as the fraction of connectivity matrices in which that particular association is observed.

3.1.6 Rank parameters selection

Input parameters of our algorithm are factorization ranks, k_1 and k_2, which we systematically examine and choose to achieve a correct reduction of dimensionality of our data. These factorization ranks capture the meaningful information that can further be decomposed into clusters.

There is no agreed-upon procedure for choosing the right factorization ranks. The most common approach, widely used in many dimension reduction problems is *cophenetic correlation coefficient*, as a quantitative measure of stability for clustering [156]. For a given factorization rank, cophenetic correlation coefficient is computed over the values of the consensus matrix, $\rho(\overline{\mathbf{C}})$. It is defined as the Pearson's correlation coefficient between the distance matrix, $1 - \overline{\mathbf{C}}$, and the matrix of cophenetic distances obtained by the linkage used in hierarchical clustering for re-ordering $\overline{\mathbf{C}}$.

Data	Matrix representation	Matrix dimension	NNZ^*
PPI	$\mathbf{L}_1^{(1)}$	3401×3401	26596
GI	$\mathbf{L}_1^{(2)}$	3090×3090	22480
Co-Ex	$\mathbf{L}_1^{(3)}$	228×228	3410
YeastNet	$\mathbf{L}_1^{(4)}$	3351×3351	21146
GDV similarity (PPI)	$\mathbf{\Lambda}_1^{(1)}$	1609×1609	93536
GDV similarity (GI)	$\mathbf{\Lambda}_1^{(2)}$	1550×1550	89434
GDV similarity (Co-Ex)	$\mathbf{\Lambda}_1^{(3)}$	122×122	2524
GDV similarity (YeastNet)	$\mathbf{\Lambda}_1^{(4)}$	1453×1453	88986
$\mathbf{L}_2$	GO semantic structure	3993×3993	15872
$\mathbf{R}_{12}$	Gene annotation	5051×3993	45782

$*$number of nonzero entries in a matrix

Table 3.1: All networks used in this study. Matrix dimensions are given before unioning genes in all data to obtain the same dimension of matrices (see Section 3.1.7). GDV matrices are of different dimension than $\mathbf{L}$- matrices because they contain only genes that are statistically significantly similar (see Section 3.1.3)

If the clustering is stable, i.e., the entries in $\overline{\mathbf{C}}$ are close to 0 or 1, then $\rho(\overline{\mathbf{C}}) \approx 1$, otherwise, if the entries are scattered between 0 and 1, $\rho(\overline{\mathbf{C}}) < 1$.

A simple generalization of this procedure applied to two types of our data (genes and GO terms) includes computation of cophenetic correlation coefficient for each of the consensus matrices, $\overline{\mathbf{C}}_g$ (for genes), $\overline{\mathbf{C}}_{GO}$ (for GO terms), and then we define the average cophenetic correlation coefficient as:

$$\rho_{avg} = \frac{\rho(\overline{\mathbf{C}}_g) + \rho(\overline{\mathbf{C}}_{GO})}{2} \tag{3.11}$$

We search for the values of, k_1 and k_2, that maximize ρ_{avg}. We do this by running our algorithm for all (k_1, k_2) pairs such that $0 < k_1, k_2 < 60$, so that we would capture the best dimensionality of our data (see Figure B.3 in Appendix B for optimal rank parameters).

3.1.7 Data sets and preprocessing

To make our study directly comparable to the competing method for reconstructing GO from network data, we run our method on the same *Saccharomyces cerevisiae* data as Dutkowski et al. [85]: PPI network from BioGRID [49], GI network from DRYGIN [68], gene Co-Ex network from SMD [59] and integrated function network, YeastNet, from [151]. For each of the these networks we construct Laplacian *constraint matrices*, $\{\mathbf{L}_1^{(1)}, \mathbf{L}_1^{(2)}, \mathbf{L}_1^{(3)}, \mathbf{L}_1^{(4)}\}$, respectively.

To apply multiplicative update rules, we make all data matrices of the same dimension: we construct them over the union of genes presented in all four data sources. The semantic structure of Gene Ontology is also taken into account in our integration algorithm. We extract all GO terms for *S. cerevisiae* and create $\mathbf{L}_2$ constraint matrix as follows. First, we construct an ontology graph using the four basic semantic types of GO relations: *is_a, part_of, regulates* and *has_part*. Then, we assign value 0.9^l to each pair of GO terms as a measure of association strength, where l is the length of directed shortest path between terms in the ontology graph. This allows us to also take into account mutual influence of hierarchically distant non-adjacent GO terms [158]. The value of 0.9 is chosen from empirical observations, as described by [70]. Finally, we construct the Laplacian constraint matrix, $\mathbf{L}_2$, by using these values of association strengths. Annotation files from GO are used to construct the binary relation matrix, $\mathbf{R}_{12}$, with entries $\mathbf{R}_{12}(i,j) = 1$ if gene i is annotated by GO term j and 0 otherwise. For each of the above mentioned biological networks, we also compute GDV similarity constraint matrices: $\{\mathbf{\Lambda}_1^{(1)}, \mathbf{\Lambda}_1^{(2)}, \mathbf{\Lambda}_1^{(3)}, \mathbf{\Lambda}_1^{(4)}\}$. As we describe in Section 3.1.3, we only consider gene pairs with statistically significant GDV similarity. All these network data are schematically represented in Figure 3.1.

3.1.8 Results and discussion

We apply our algorithm to identify new GO term relations and annotate proteins with existing GO terms by integrating multiple independent network sources given in the Table 3.1. We find that the optimal rank parameters k_1 and k_2 are 58 and 56, respectively (see Fig. B.3 in Appendix B). We examine the contribution of each data source to the integration model.

Contribution of networks to the integration model

We estimate the influence of each network on our integration model by comparing the quality of the initial model (consisting of four networks and their corresponding GDV similarity matrices) with the quality of the model with one data source removed from the initial set. Models are evaluated through residual sum of squares (RSS), $RSS(\mathbf{R}_{12}) = \sum_{ij}[\mathbf{R}_{12}(i,j) - (\mathbf{G}_1\mathbf{S}_{12}\mathbf{G}_2^T)(i,j)]^2$, and explained variance, $Evar(\mathbf{R}_{12}) = 1 - RSS(\mathbf{R}_{12})/\sum_{ij}[\mathbf{R}_{12}(i,j)]^2$, that measure the performance of the matrix factorization algorithm and its ability to accurately reproduce the gene-GO term relation matrix. Low values of RSS and high values of Evar indicate better quality of the model [159].

We find that with the removal of each of the four data sources (a network along with its corresponding GDV similarity matrix) the value of RSS increases, while the

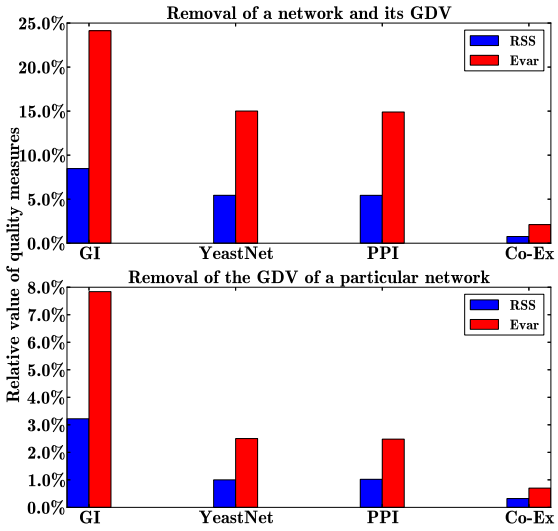


Figure 3.2: Relative contribution of each data source to the integration model measured by residual sum of squares (RSS), blue, and explained variance (Evar), red. The top panel shows the relative changes in RSS and Evar with the removal of a particular network and its corresponding GDV similarity matrix. The bottom panel shows the same measures, but only with removal of GDV similarity matrices. The figure is taken from Gligorijević et al. [26].

value of Evar decreases, implying that each data source contributes to the quality of the model. Relative increase of RSS and relative decrease of Evar (with respect to the initial model containing all the data), computed by removing a particular network along with its corresponding GDV similarity matrix, are shown in the top panel of Figure 3.2. We find that the largest model degradation is achieved with the removal of GI network and its corresponding GDV similarity matrix. A similar result was reported by [70]: they found GIs to be the most informative data source in prediction of disease-disease associations. Exclusion of the gene co-expression network and its corresponding GDV similarity matrix results in the smallest changes in RSS and Evar indicating that co-expression data contribute the least to the quality of the model.

To examine the contribution of GDV similarities to our model, we conduct the same experiment by removing only the GDV similarity matrix of each of the biological networks from the initial data set. The results are shown in the bottom panel of Figure 3.2. We see that GDV similarities contribute to the quality of the models. The smallest contribution to the model, a relative increase of 0.32% in RSS, is that of the gene co-expression network. Also, we examine contributions of all pairs of the four networks. We confirm the observation of [85] that a combination of YeastNet and co-expression network contributes the least to the to the quality of the model ($RSS = 0.8\%$, $Evar = 1\%$).

Recovering existing knowledge

To evaluate the performance of our methodology in reproducing GO term associations, we look at the overlap between GO term associations obtain from connectivity matrix, $\hat{\mathbf{C}}_{GO}$ and associations in existing GO hierarchy. We find that on average 92% of highly associated GO terms are directly connected via semantic relations in GO. These cluster-induced GO-term relations are confirmed to be statistically significant ($p \leq 0.01$, computed as explained in Section 3.1.5). A slightly lower score of 90% is achieved when considering only the network data without GDV similarity matrices, indicating that graphlet similarity matrices contribute to capturing relations which would otherwise be missed. Furthermore, we examine the robustness of this result to the removal of particular data sets. Surprisingly, we find that omission of GDV similarity matrix of gene co-expression network contributes the most to the predictive performance of our algorithm, leading to the maximum of 96% of recovered GO terms. Hence, inclusion of GDV similarity of gene co-expression network introduces noise into the integration procedure, wrongly guiding the clustering process which in turn results in lower prediction performance. This is a consequence of the random GDV similarity distribution over all genes in the gene Co-Ex network (see Fig. B.2 in Appendix B). Given that inclusion of GDV similarity matrix of the Co-Ex network impairs the predictive performance of our algorithm and since we shown that its exclusion makes minimal effect on the quality of the model we discard that data source from further analysis.

Surprisingly, recovering 96% of GO terms that are directly related in GO by *is_a*, *part_of*, *regulates*, and *has_part* associations, indicates that entire Gene Ontology could, in principle, be reconstructed solely from topologies of molecular interaction networks. Reporting this statistic is consistent with what previous studies using a similar methodology reported [70]. Our set of 96% of GO terms contains 78% of all relations currently present in GO. To our knowledge, since a large part of GO is sequence derived, this is the first conformation that network topology and sequence carry very similar biological information.

To further validate the performance of our methodology in reconstruction of GO terms, we use the gene-GO term relation matrix, reconstructed from matrix factors: $\hat{\mathbf{R}}_{12} = \mathbf{G}_1\mathbf{S}_{12}\mathbf{G}_2^T$. Its entries indicate the annotation strength of a gene, i, related to a GO term j, with $\hat{\mathbf{R}}_{12}(i,j) = 0$ denoting absence of annotation, while $\hat{\mathbf{R}}_{12}(i,j) = 1$ denoting the highest confidence of annotation. We define GO term j^* as a candidate to annotate gene i if the association score $\hat{\mathbf{R}}_{12}(i,j^*)$ is larger that the mean of association scores over all known annotations of gene i. To identify only high confidence gene-GO term predictions we pick j^* that are in the top 5% of largest association scores between GO term j^* and all other genes. As before, we

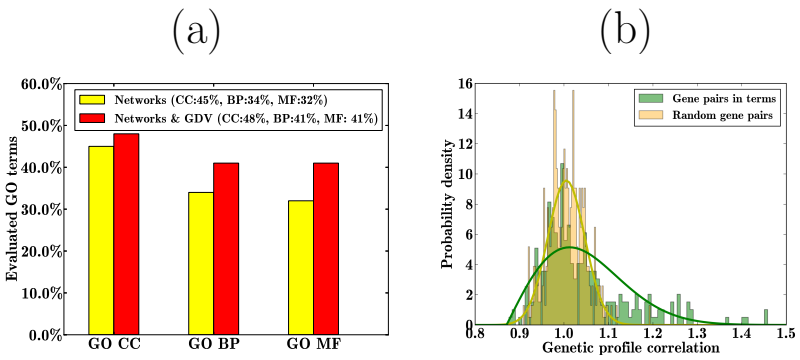


Figure 3.3: (a) The fraction of GO terms in each of CC (cellular component) BP (biological process) and MF (molecular function) obtained from entries of reconstructed gene-GO term relationship matrix obtained with and without GDV similarities (denoted in red and yellow color respectively). (b) Distribution of correlations of genetic interaction profiles among predicted genes associated to GO terms plotted against distributions of randomly selected gene pairs. Value of correlation, presented here, is shifted in the positive range: [0,2]. The figure is taken from Gligorijević et al. [26].

run our algorithm with an without GDV similarities (we exclude GDV similarities of Co-Ex network, for reasons presented above).

We compute the percentage of reproduced, high confidence GO terms for cellular component (CC), biological process (BF) and molecular function (MF) separately. The results are shown in Figure 3.3 (panel A). Better results are achieved when GDV similarity matrices are included in the prediction model. Specifically, we capture 41% of BP terms, 41% of MF terms and 48% of CC terms. The BP and MF results outperforms those of Dutkowski et al. [85], while they achieve a higher percentage of reproduced GO terms in CC.

Validating predictions

Among all the statistically significant ($p \leq 0.01$) GO term association predictions, we find 132 not presented in GO[2]. To further increase confidence, we extract these associations from clusters with fewer than three GO terms that are stable over multiple factorization runs. We find that 14 out of the 132 associations are between GO terms that have high semantic similarity and also confirm that additional 31 associations agree with predictions of [85]. For example, our approach predicts term GO:0035267 (NuA4 histone acetyltransferase complex) as a parent of GO:0032777 (Piccolo NuA4 histone acetyltransferase complex), which was also reported by [85] and submitted to the GO Consortium for inclusion into the ontology. We further

[2] see online supplementary table at https://www.doc.ic.ac.uk/~vgligori/docs/go_predictions.xlsx

perform literature curation to validate the remaining predicted GO associations. We find literature support for another 13 of them[2]. Hence, we validate 58 out of 132 of our predictions.

Our approach not only identifies novel GO term association, but it also makes highly reliable predictions for new gene-GO term relations. We predict new functional annotation of 972 genes[2]. Highly reliable predictions are those with association strength in the top 5%, as described in the previous section. For instance, we predict three genes, YDR101C, YDR49C and YNL132W, to be involved in ribosomal subunit biogenesis (GO:0042273) and find that the same functional prediction was previously reported through different approaches by Chen and Xu [142] and Joshi et al. [160]. To validate the 972 predicted annotations we use the new, unpublished, full set of yeast's genetic interaction profiles from Boone Lab [161]. The data consist of Pearson's correlation coefficients of genetic profiles between gene pairs. We create the distribution of these correlations between newly annotated gene pairs for which we predict GO annotations. We compare this distribution of genetic profile correlations between the same number randomly sampled pairs of genes (we sampled multiple times and got consistent results). We observe higher correlations for predicted gene pairs than for random pairs (Fig. 3.3 (panel B)). Moreover, using *two-sample Kolmogorov-Smirnov (KS) test* we show that these two distributions significantly differ (KS statistics, $D = 0.2$ and p-value, $p = 1.5 \times 10^{-6}$). Thus, these results are highly consistent with our predictions of new annotations. This validate our predicted GO annotations. Even though genetic interaction profiling analysis provides evidence that our algorithm is able to successfully predict new gene functions, additional biological validation would be needed for better understanding of these newly assigned functions.

3.1.9 Conclusions

We introduce a method for reconstruction of GO that is based on integrating solely the topology of biological networks. It captures 96% of the existing GO term relations and is capable of successfully identifying additional GO term associations as well as predicting gene annotations. Our method is general in the sense that it can integrate any heterogeneous systems-level interaction data. Therefore, it can easily be extended with new data that could consequently enhance the model's predictive performance. This work suggests that the entire Gene Ontology could be reconstructed from molecular interaction networks.

3.2 Data integration for precision medicine

According to Cancer Research UK, cancer is a leading cause of death accounting for more than one in four of all deaths in 2011. The recent advances in experimental technologies in cancer research have resulted in the accumulation of large amounts of patient-specific data sets, which provide complementary information on the same cancer type.

We introduce a versatile data fusion (integration) framework that can effectively integrate somatic mutation data, molecular interactions and drug chemical data to address three key challenges in cancer research: *stratification of patients* into groups having different clinical outcomes, *prediction of driver genes* whose mutations trigger the onset and development of cancers, and *repurposing of drugs* treating particular cancer patient groups. Our proposed framework is based on GNMTF technique introduced in the previous sections. We apply our framework on ovarian cancer data to simultaneously cluster patients, genes and drugs by utilising *all* data sets.

We demonstrate superior performance of our method over the state-of-the-art method, Network-based Stratification, in identifying three patient subgroups that have significant differences in survival outcomes and that are in good agreement with other clinical data. Also, we identify potential new driver genes that we obtain by analysing the gene clusters enriched in known drivers of ovarian cancer progression. We validated the top scoring genes identified as new drivers through database search and biomedical literature curation. Finally, we identify potential candidate drugs for repurposing that could be used in treatment of the identified patient subgroups by targeting their mutated gene products. We validated a large percentage of our drug-target predictions by using other databases and through literature curation.

3.2.1 Introduction

Cancer is a leading cause of morbidity worldwide. It is a complex genetic disease in which the genomes of normal cells accumulate somatic mutations and other alterations that are eventually perturbing vital cellular functions. Recent advances in DNA sequencing technologies have enabled identification of somatic mutations across tumor genomes and exomes of individual patients [162, 163]. These somatic mutations provide a new and rich source of data for addressing many challenges in cancer research, such as identifying driver genes (i.e., genes whose mutations lead progression of oncogenesis), stratifying patients into biologically meaningful classes with different clinical outcomes and creating new opportunities for development of successful personalized treatment strategies [164]. Cancer is also a highly heteroge-

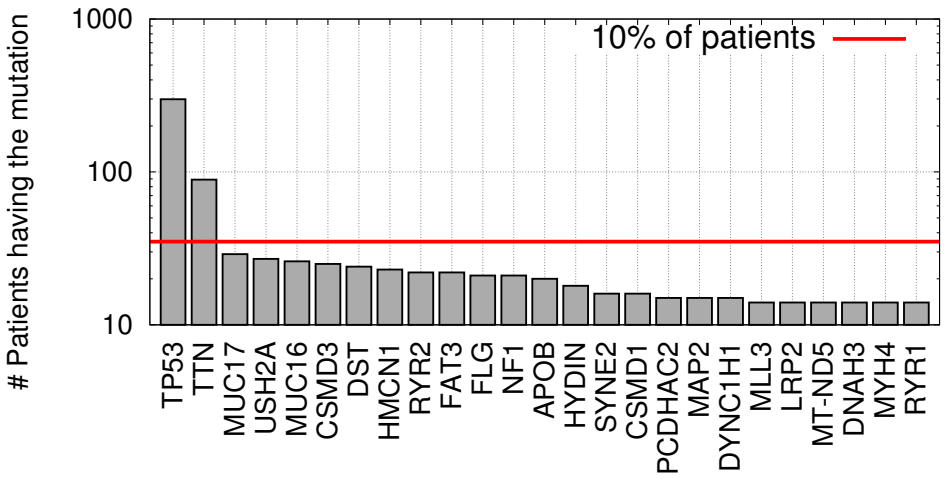


Figure 3.4: The top 25 most frequently mutated genes in the 535 serous ovarian cancer patients from TCGA. For each gene (x-axis), we report the number of patients having the gene mutated (y-axis, log-scaled). The red line corresponds to 10% of the patients. Only two genes, TP53 and TTN, are mutated in more than 10% of the patients.

neous disease with large genetic diversity even between tumors of the same cancer type. Namely, two clinically identical tumors rarely have a large set of common mutated genes. Moreover, very few genes are frequently mutated across tumor samples (see Fig. 3.4 for top 25 genes most frequently mutated in ovarian cancer). This makes the use of somatic mutations for identification of driver genes, as well as for patient stratification into subtypes, much harder [162, 165, 166]. However, despite this genetic diversity between tumor samples, the perturbed pathways are often similar [162]. Therefore, integration of somatic mutations with other genomic data, such as with molecular networks that contain pathways, is a promising direction for addressing these problems.

Development of computational methodologies that can efficiently integrate genome-scale molecular information and address the above mentioned challenges in cancer research is of foremost importance. A majority of previous studies do not utilise data on somatic mutations, but instead, they are mainly based on mRNA expression and methylation data. Because of noisiness of these data, the patient stratification studies for cancer types often do not produce patient subgroups that agree well with any clinical or survival data [167]. Therefore, a recent study proposed the use of somatic mutation data in combination with biological networks as a new source of information for tumor stratification [166]. However, the proposed methodology cannot account for additional data types (e.g., drug data) and cannot be used for identifying novel driver genes, nor for predicting a personalised therapy. Moreover, previous data integration methods can only be used for either cancer patient stratification

[166], driver gene prediction [168] or drug repurposing [169].

Here, we present a versatile patient-specific data integration (fusion) methodology capable of: 1) uncovering patient subgroups (stratification) with prognostic survival outcome, 2) predicting novel driver genes and 3) repurposing drugs, i.e., predicting new candidate drugs for targeting mutated gene products in individual patients and that can be used in treatment of identified patient subgroups. To our knowledge, this is the first method that can address all three challenges simultaneously.

We use GMTF to integrate somatic mutation profile (SMP) data of serous ovarian cancer patients from TCGA [165] with molecular networks (MNs) from BioGRID[170] and KEGG [171], drug-target interaction (DTI) and drug chemical similarity (DCS) data from DrugBank [172] (detailed in Sec. 3.2.3). We perform consensus clustering by using GNMTF to simultaneously cluster patients, genes and drugs based on the evidence from *all* data sets. First, we stratify patients into three groups that we assess by using clinical data. We show significant difference in survival outcomes between these groups, as well as a good agreement with other clinical data. Second, from clusters of genes, we identify those enriched in known driver mutations; we postulate genes strongly related to known driver genes in these clusters as potential drivers genes, i.e., genes responsible for ovarian cancer progression. Finally, we use the matrix completion property of NMTF to predict new drug-target relations and to identify new drug candidates that could be used for repurposing and treatment of identified ovarian cancer patient groups. Furthermore, we evaluate the influence of all combinations of data sets onto the accuracy of drug-target predictions by performing a 5-fold cross validation. We shown that the highest accuracy is achieved when all data sets are taken into account, proving the utility of integrating all considered data sets (detailed in Sec. 3.2.4).

3.2.2 Methods

Patient-specific data fusion framework

We consider there different data sets: patients, genes and drugs. Patients and genes are related to each other by somatic mutation profiles (SMPs), constructed for n_1 patients over n_2 genes and encoded in high-dimensional relation matrix, $\mathbf{R}_{12}^{n_1 \times n_2}$. Its entries are binary values, with $\mathbf{R}_{12}(p, g) = 1$ if gene g is found to be mutated in patient p, and zero otherwise. Genes and drugs are related to each other according to drug-target interactions (DTIs). DTIs between n_2 genes and n_3 drugs are encoded in relation matrix, $\mathbf{R}_{23}^{n_2 \times n_3}$. Its entries are also binary values, with $\mathbf{R}_{23}(g, d) = 1$, if

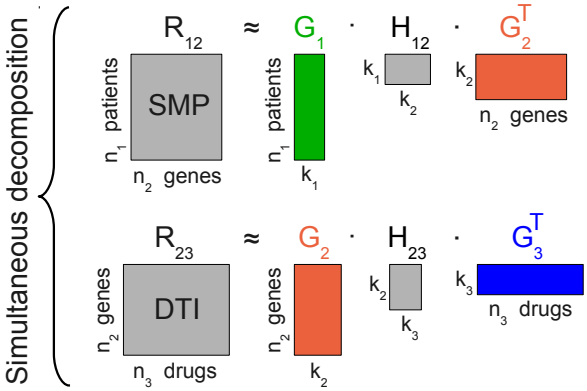


Figure 3.5: Our patient-specific data integration framework.

the product of gene g is a target of drug d and zero otherwise. See Section 3.2.3 and Figure 3.6 for details of construction of the relation matrices and for an illustration of these data sets.

We use NMTF to simultaneously decompose both relation matrices into a product of three non-negative low-dimensional latent feature matrices as follows: $\mathbf{R}_{12} \approx \mathbf{G}_1\mathbf{H}_{12}\mathbf{G}_2^T$. and $\mathbf{R}_{23} \approx \mathbf{G}_2\mathbf{H}_{23}\mathbf{G}_3^T$. The low dimensional matrices can be obtained by solving the following optimisation problem:

$$\min_{\mathbf{G}_i \geq 0, 1 \leq i \leq 3} J = \min_{\mathbf{G}_i \geq 0, 1 \leq i \leq 3} \Big(\parallel \mathbf{R}_{12} - \mathbf{G}_1\mathbf{H}_{12}\mathbf{G}_2^T \parallel^2 + \parallel \mathbf{R}_{23} - \mathbf{G}_2\mathbf{H}_{23}\mathbf{G}_3^T \parallel^2 \Big),$$

where J is the objective function. The non-negativity constraints imposed on $\mathbf{G}_i$ matrices, for $1 \leq i \leq 3$, provide easier interpretation of their values in the clustering assignment. As we have seen in the previous section (Sec. 3.1) many of the data types are characterised by additional, internal connectivity structure represented by graphs (networks). In this study, genes are connected by molecular networks (MNs), while drugs are connected based on the similarity of their chemical structures, i.e., drug chemical similarity (DCS) network (illustrated in Fig. 3.6). We incorporate these networks (MN and DSC) into the above objective function by adding two regularisation terms to constrain the construction of $\mathbf{G}_2$ and $\mathbf{G}_3$ matrices. Thus, we arrive again to GNMTF technique. Namely, the aim is to enforce two interacting genes to belong to the same cluster (similarly with drugs) and a violation of these constrains results in penalties to our objective function. Hence, the final objective function has the following form:

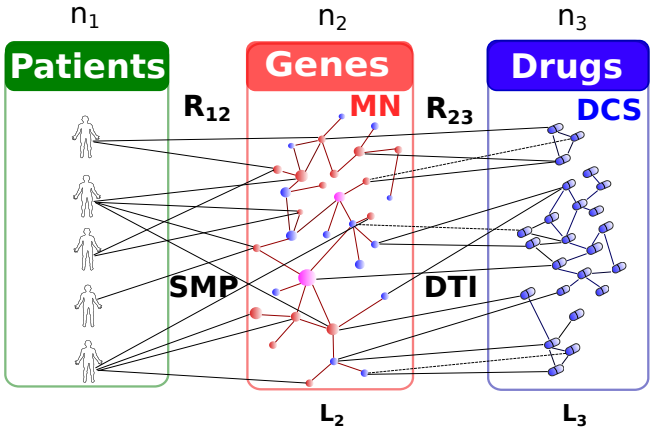


Figure 3.6: Schematic illustration of data sets used in this study. Three types of objects (nodes) are represented: n_1 *ovarian cancer patients* (in green), n_2 *genes* (in red) and n_3 *drugs* (in blue), and their *inta-type* and *inter-type* relations. The figure is taken from Gligorijević et al. [139].

$$\min_{\mathbf{G}_i \geq 0, 1 \leq i \leq 3} J = \min_{\mathbf{G}_i \geq 0, 1 \leq i \leq 3} \big[\parallel \mathbf{R}_{12} - \mathbf{G}_1 \mathbf{H}_{12} \mathbf{G}_2^T \parallel_F^2 + \parallel \mathbf{R}_{23} - \mathbf{G}_2 \mathbf{H}_{23} \mathbf{G}_3^T \parallel_F^2 + + tr(\mathbf{G}_2^T \mathbf{L}_2 \mathbf{G}_2) + tr(\mathbf{G}_3^T \mathbf{L}_3 \mathbf{G}_3) \big] \tag{3.12}$$

where, $\mathbf{L}_2$ and $\mathbf{L}_3$ are graph Laplacians of MN and DCS networks, respectively.

The key idea of our GNMTF-based data fusion approach is in sharing low-dimensional matrix $\mathbf{G}_2$ whilst simultaneously learning from (i.e., decomposing) relation matrices, $\mathbf{R}_{12}$ and $\mathbf{R}_{23}$ (see Fig. 3.5)[3]. Such decomposition accounts for collective influence of all data sets (along with molecular and chemical constraints effectively integrated within the same framework) onto the resulting clustering of patients, genes and drugs. This approach corresponds to the *intermediate* data fusion in which the structure of the data is preserved during the model inference. Such an approach has been shown to result in the best accuracy among all data fusion approaches [70].

Minimisation of the objective function, J, is done by *multiplicative update rules* used to compute all low-dimensional matrices; under these multiplicative rules, the objective function is non-increasing [137]. The minimisation starts by randomly initialising $\mathbf{G}_i$ matrices for $1 \leq i \leq 3$ and then iteratively updating their values until the convergence criterion is reached. In all our runs, we use *Random Acol* initialisation strategy [129] and the convergence criterion is reached when $\frac{|J_{n+1} - J_n|}{|J_n|} <$

[3]Note that objective function in Equation 3.12 is more general than objective function in Equation 3.7 as it contains also the decomposition term of another relation matrix, i.e., $\mathbf{R}_{23}$, whose factorization is coupled with the factorization of $\mathbf{R}_{12}$ matrix trough $\mathbf{G}_2$ matrix.

10^{-5}. The multiplicative update rules, their derivation and proof of convergence can be found in Wang et al. [137].

Co-clustering of patients, genes and drugs. Matrices $\mathbf{G}_1^{n_1 \times k_1}$, $\mathbf{G}_2^{n_2 \times k_2}$ and $\mathbf{G}_3^{n_3 \times k_3}$ from Equation 3.12 above are *soft cluster membership indicator* matrices for patients, genes and drugs, respectively; based on their entries, n_1 patients are assigned to k_1 patient clusters, n_2 genes are assigned to k_2 gene clusters and n_3 drugs are assigned to k_3 drug clusters, respectively. In particular, following the *hard clustering* procedure of Brunet et al. [156], matrix $\mathbf{G}_1^{n_1 \times k_1}$, with rows representing patients and columns representing clusters, is used to place patient p into cluster k if $\mathbf{G}_1(p, k)$ is the largest entry in row p. This assignment procedure results in the binary connectivity matrix for patients, $\mathbf{C}_1^{n_1 \times n_1}$, with entry $\mathbf{C}_1(p_1, p_2) = 1$ if patients p_1 and p_2 belong to the same cluster and $\mathbf{C}_1(p_1, p_2) = 0$ otherwise. We apply this procedure for all cluster membership indicator matrices. The number of clusters (also called *rank parameters*) for each data set are chosen to be $k_1 \ll n_1$, $k_2 \ll n_2$ and $k_3 \ll n_3$, which provides dimensionality reduction of the relation matrices. Matrices $\mathbf{H}_{12}^{k_1 \times k_2}$ and $\mathbf{H}_{23}^{k_2 \times k_3}$ in Equation 3.12 above represent compressed, low-dimensional versions of $\mathbf{R}_{12}$ and $\mathbf{R}_{23}$, respectively.

An important step in our methodology is estimating rank parameters, which are the numbers of clusters of patients, genes and drugs, k_1, k_2 and k_3, respectively. Similar to the problem of protein function prediction, these parameters need to be known before factorisation is performed. We use the same procedure for estimating rank parameters as described in Section 3.1 [156, 173]. We take the values of parameters for which the most stable clustering is achieved. In particular, multiplicative update rules converge to a different solution in each run, depending on the random matrix initializations (i.e., initial clustering assignment given by the initial values in matrices $\mathbf{G}_i$, $1 \leq i \leq 3$). For example, if a clustering of patients into k_1 classes is stable, we expect small variations in the assignment to clusters from run to run. To measure this, we perform multiple factorisation runs with the same values of rank parameters. Each time, a connectivity matrix is computed (e.g., $\mathbf{C}_1$ for patients); based on these, an averaged connectivity matrix (also called the *consensus* matrix) over all runs is computed, $\hat{\mathbf{C}}_1$. If the clustering is stable, then the entries in $\mathbf{C}_1$ (also referred to as the *cluster association scores*) will be either close to zero, or close to one. Otherwise, the entries will be scattered in the interval $[0, 1]$. We use the *dispersion* coefficient, $\rho_{k_1}(\hat{\mathbf{C}}_1)$, introduced by Kim and Park [173], as a measure of cluster stability. The values of the dispersion coefficient range in $0 \leq \rho_{k_1}(\hat{\mathbf{C}}_1) \leq 1$, where 1 denotes a stable clustering. In our study, for each rank parameter, we perform a grid search in intervals of 1 for $1 \leq k_1 \leq 5$, $5 \leq k_2 \leq 30$ and $5 \leq k_3 \leq 30$, and compute dispersion coefficients, $\rho_{k_1}(\hat{\mathbf{C}}_1)$, $\rho_{k_2}(\hat{\mathbf{C}}_2)$ and $\rho_{k_3}(\hat{\mathbf{C}}_3)$ for patients, genes and drugs,

respectively. We choose the values for k_1, k_2 and k_3 for which dispersion coefficients are of the highest values.

Matrix completion property. In addition to co-clustering of patients, genes and drugs, we model the existing and predict new drug-target interactions by using the *matrix completion* property of NMTF. Namely, after obtaining low-dimensional matrices, the reconstructed drug-target matrix, $\hat{\mathbf{R}}_{23} \approx \mathbf{G}_2\mathbf{H}_{23}\mathbf{G}_3^T$, is more dense than the initial matrix, $\mathbf{R}_{23}$, and it can be used for extracting new, unobserved drug-target relations and therefore, finding new drug candidates for repurposing.

Drug repurposing, patient stratification and driver gene prediction

Drug repurposing. We use the reconstructed drug-target relation matrix, $\hat{\mathbf{R}}_{23}$, to extract new, previously, unobserved drug-gene interactions and to postulate new candidates for drug repurposing in the treatment of ovarian cancer patients. We apply a combination of row-centric and column-centric rules to extract new, strongly associated drug-gene pairs [88]. Namely, a drug-gene pair, (d, g), is considered to be predicted, if the estimated association score, $\hat{\mathbf{R}}_{23}(g, d)$, is greater than the mean association score of all relations of gene g, as well as greater than the mean association score of all relations of drug d.

Patient stratification. We stratify ovarian cancer patients into groups, according to the consensus matrix, $\hat{\mathbf{C}}_1$. We use the approach of Brunet et al. [156]: we use the off-diagonal entries of $\hat{\mathbf{C}}_1$ as a measure of patient similarity and apply average linkage hierarchical clustering to group patients into k_1 classes. Results and validations are shown in Section 3.2.4 below.

Cancer driver gene prediction. Similar to the patient consensus matrix, we use the gene consensus matrix, $\hat{\mathbf{C}}_2$, to extract gene clusters and identify those that are enriched in mutations and known driver genes by using *the standard model sampling without replacement test (i.e., hypergeometric test).* In clusters that are enriched in known drivers, we identify genes that are highly associated with known driver genes based on the clustering association scores from the gene consensus matrix. We postulate that these genes are new driver genes for ovarian cancer. Results and validations are presented in Section 3.2.4 below.

Clinical and biological validation of results. For all patients, we also downloaded clinical follow up data from TCGA database, including the overall patients' survival information (days to the last follow-up and vital status), age, tumor grade, size and tumor position. We used these data to assess the clinical relevance of the patient clusters that we obtain after data fusion. We used Kaplan-Meier survival curves, as well as the log-rank p-value, to measure the significance of the difference in survival profiles between different patient clusters. The log-rank p-value measures the probability of the null hypothesis that patients in each cluster are drawn from the same underlying survival distribution [174]. From TCGA database, we also retrieved a list of 83 known ovarian cancer driver genes, out of which 76 are present in our set of mutated patient genes. We use this set of genes to assess gene clusters obtained after fusion and to identify clusters enriched in drivers.

To validate the predicted drug-target interactions, we use two different drug-target databases: Comparative Toxicogenomics Database (CTD) [175] and Manually Annotated Targets and Drugs Online Resource (MATADOR) [176]. Both databases assemble DTIs by automated text-mining followed by manual curation. We choose these databases because of their complementarity to DrugBank database. For each predicted drug-target pair, we report whether or not it appears in any of the two databases.

3.2.3 Datasets, preprocessing and matrix construction

We downloaded high-grade serous ovarian cancer somatic mutation data from TCGA data portal [165] on the 2^{nd} of July 2015. We only consider data generated by using Illumina GAIIx platform, having the largest number of patients. Following the same procedure for data filtering as in Hofree et al. [166], we retain only the patients with more that 10 mutated genes. This results in $n_1 = 353$ serous ovarian cancer patients with mutations in the total of $11,148$ genes. Mutated genes are mapped onto the Molecular Network (MN) that we obtain by merging three different biological networks: PPI and GI network from BioGRID database (version 3.4.126) [170], and MI network from KEGG database [171] (see Table 2.1 for network abbreviations). This results in MN of $236,751$ interactions among $n_2 = 19,118$ genes (mutated and normal). We represent these interactions by Laplacian matrix, $\mathbf{L}_2^{n_2 \times n_2}$, computed as: $\mathbf{L}_2 = \mathbf{D}_2 - \mathbf{A}_2$, where $\mathbf{A}_2$ is the adjacency matrix of MN and $\mathbf{D}_2$ is the diagonal degree matrix of MN (i.e., whose entries on the diagonal are row sums of $\mathbf{A}_2$ and all other entries in $\mathbf{D}_2$ are zeros). For each patient, we create an n_2- long binary $(0, 1)$ somatic mutation profile (SMP) vector, where "1" indicates the existence of a mutated gene in the patient and all other entries are "0". These mutation profiles for all n_1 patients are captured in a binary relation matrix $\mathbf{R}_{12}^{n_1 \times n_2}$ consisting of these SMP vectors.

Due to the sparsity of matrix $\mathbf{R}_{12}$, we apply a network propagation technique as the preprocessing step to smooth the patient profiles, by spreading the influence of each mutation over its neighbours in MN network. We use the network propagation method proposed by Vanunu et al. [177], based on which the new patient-gene relation matrix is computed iteratively as follows: $\mathbf{R}_{12}^{t+1} = \alpha \mathbf{R}_{12}^{t} \bar{\mathbf{A}}_2 + (1-\alpha)\mathbf{R}_{12}^{0}$, where $\bar{\mathbf{A}}_2$ is the normalised adjacency matrix of MN network computed as $\bar{\mathbf{A}}_2 = \mathbf{A}_2\mathbf{D}_2^{-1}$, $\mathbf{R}_{12}^{0} = \mathbf{R}_{12}$ is the initial patient-gene matrix and α is a tuning parameter that controls the distance of diffusion through MN network. In all our runs, we set $\alpha = 0.6$ (as it produced the best results), and we took the final network-smoothed, patient-gene matrix (after convergence, $|\mathbf{R}_{12}^{t+1} - \mathbf{R}_{12}^{t}| < 10^{-6}$, is achieved) as input to GNMTF. This preprocessing step has been shown to lead to much better and more robust patient stratification results in previous studies [166], hence we use it as well.

Drug-target interactions (DTIs) are downloaded from DrugBank database (version 4.3) [172]. We retrieved $n_3 = 6,620$ drugs (FDA-approved and experimental) targeting $1,385$ genes in MN. These interactions are captured by DTI binary relation matrix, $\mathbf{R}_{23}^{n_2 \times n_3}$. SMILES chemical representation of the n_3 drugs are also retrieved from DrugBank database. The two-dimensional chemical similarity between drugs are computed by using Tanimoto similarity coefficient [77]. We retain only the top 5% most similar drug pairs, which results in $1,069,393$ links in the drug chemical similarity (DCS) network. We represent these links by Laplacian matrix, $\mathbf{L}_3^{n_3 \times n_3}$ (computed in the same way as for MN network, described above).

3.2.4 Results

Patient stratification

We perform the consensus clustering, as described in Section 3.2.2, with 20 different random initialisations (initial cluster assignment) of GNMTF and compute the consensus matrices of patients, genes and drugs. We observe that rank parameters of $k_1 = 3$ (the number of patient clusters), $k_2 = 25$ (the number of gene clusters) and $k_3 = 20$ (the number of drug clusters), lead to the most stable clustering (with the largest dispersion coefficients: $\rho_{k_1=3}(\hat{\mathbf{C}}_1) = 0.56$, $\rho_{k_2=25}(\hat{\mathbf{C}}_2) = 0.91$ and $\rho_{k_3=20}(\hat{\mathbf{C}}_3) = 0.88$).

To assess the prognostic capabilities of our patient-specific data fusion approach on ovarian cancer patients, we perform clinical validation of the three obtained patient clusters. The Kaplan-Meier survival curves, shown in Fig.3.7 (A), reveal the low-survival group (*Cluster 2*) with 56% of death cases and the good outcome

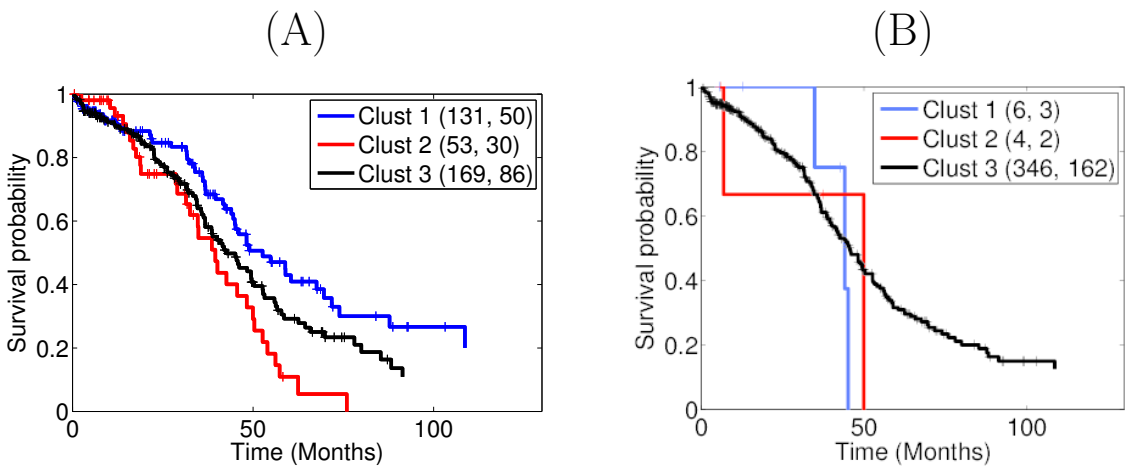


Figure 3.7: Kaplan-Meier survival curves for 3 different patient groups produces by GNMTF (A) and NBS (B). The total number of patients and the number of deceased patients for each cluster are shown in the legend, the first and the second number in brackets, respectively. The figure is taken from Gligorijević et al. [139].

group (*Cluster 1*) with 38% of death cases. We observe that the identified clusters are highly discriminative with the log-rank p-value of 5.3×10^{-3}. The same number of clusters has been also reported in previous studies done on somatic mutation and molecular interaction data [166], and also in study done only on miRNA expression data [165]. Furthermore, the identified clusters display a good agreement with the median age of patients in clusters, with *Cluster 2* having the oldest patients. In addition, *Cluster 2* has the largest number of patients with abnormal growth of tissue (tumor), 78%, as compared to *Cluster 1* with 60% of such patients.

We compare the performance of our method with the state-of-the-art somatic mutation-based stratification method called Network-based Stratification (NBS)[166]. NBS takes as input a patient-gene post-smoothing relation matrix and a molecular network matrix. We apply it on the same set of data described in Section 3.2.3, excluding drug data, which only our framework can take into account. We test NBS for different numbers of patient clusters (i.e., $k \in \{2, 3, 4, 5\}$) and compute the Kaplan-Meier survival curves [174] for the obtained patient clusters. We compare the survivability results of NBS with our method with the same number of patient clusters (Fig. 3.7 (A, B)). Unlike our method, which can produce clusters with significantly different survival outcomes (i.e., p-val $= 5.3 \times 10^{-3}$), NBS cannot (p-val ≥ 0.74 for all $k \in \{2, 3, 4, 5\}$). Thus, our framework is the only one able to extract personalised knowledge from somatic mutation profiles.

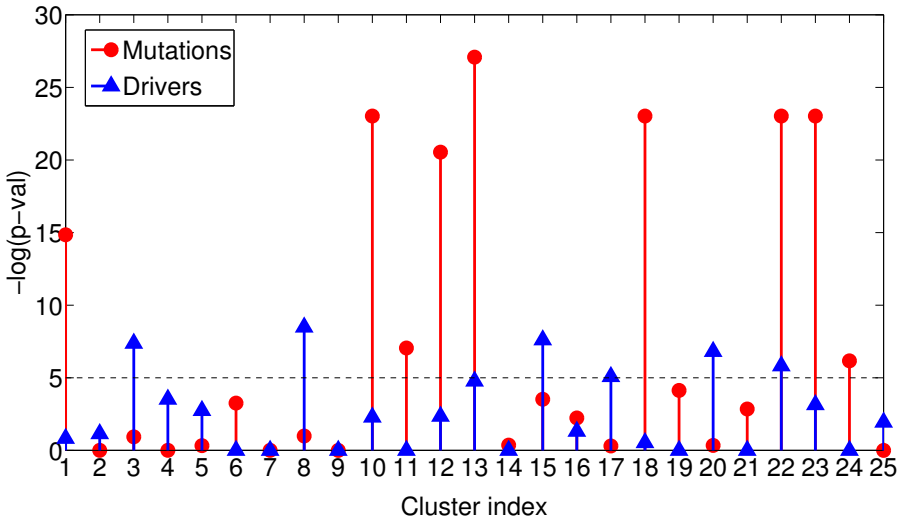


Figure 3.8: Clusters enriched in mutated (red) and driver (blue) genes. For each cluster, $-log(pval)$ is plotted as a measure for enrichment (y-axis). The figure is taken from Gligorijević et al. [139].

Identification of driver genes

We performed biological assessment of the $k_2 = 25$ gene clusters that we obtain from the gene consensus matrix, $\hat{\mathbf{C}}_2$. We identify 9 gene clusters that are significantly enriched in mutations and 5 gene clusters that are significantly enriched in known drivers ($p - val \leq 0.05$, see Fig. 3.8). Out of these clusters, cluster number 8 has the largest number of driver genes (26) and the highest enrichment in driver genes (with $p - val = 2.06 \times 10^{-4}$). To identify new driver genes, we further analyse this cluster as follows: first, based on the cluster association scores in the gene consensus matrix, we extract the mutated genes that are strongly associated with the known driver genes. In particular, we focus only on genes associated with the known driver genes with the cluster association score ≥ 0.9 (as explained below). That is, out of 20 restarts of GNMTF, we extract genes that appear 18 times in cluster 8 with other driver genes. Then, for each of these genes, we compute the average cluster score based on its associations with all driver genes. We provide the list of the top 20 genes (out of 809 predicted drivers in total) that we postulate as new driver genes of ovarian cancer progression and we sort it according to the average cluster association score, as shown in Table 3.2. This procedure is motivated by the observation that out of the 76 known driver genes, 67 of them are strongly related (with cluster association score ≥ 0.9) among themselves through all gene clusters.

We assess our predicted driver genes against two cancer driver gene databases, COSMIC database Cancer Gene Census [178] and IntOGen [179], as well as against a database of putative cancer driver genes, the Candidate Cancer Gene Database

(CCGD) [180]. Our results show that $\sim 40\%$ of our 809 predicted driver genes (with scores ≥ 0.9) have either been already proposed as drivers (in CCGD), or validated by experts (Census, or IntOGen). The list of our 20 top-scoring predicted cancer driver genes is presented in Table 3.2. Also, we investigated the literature to assess the relevance of our two top-scoring predictions that are not found in other databases and found evidence that they are biologically relevant. Our top-scoring cancer driver gene prediction is ADAM32, which is strongly clustered with driver gene BMPR2 (Table 3.2). The association between the two genes is biologically relevant, because both are involved with transforming growth factors (TGFs). Our prediction of ADAM32 as a cancer driver gene is also relevant, because ADAM genes are known to be responsible for cancer cell proliferation and progression [181]. The second best prediction is REG1P (from the REG family of proteins), which is strongly clustered with driver gene CLASP2. Our prediction of REG1P as a cancer driver gene is also relevant, because the REG family plays different roles in proliferation, migration, and anti-apoptosis through activating different signalling pathways; their dis-regulation is closely associated with cancer and REG proteins have been proposed as markers for prognosis of cancers [182].

Drug-target interaction prediction

To demonstrate the predictive power of our data fusion approach and to assess the contribution of each data set on the drug-target interaction prediction, we perform a 5-fold cross validation for each combination of the datasets shown in Fig. 3.6. In all our experiments, true positives are correctly predicted DTIs, while false positives are predicted DTIs that are not present in the initial data set.

We compute average Area Under the Receiver Operator Characteristic (ROC) and Precision-Recall (PR) curves (over 20 repetitions) to evaluate the performance of our methodology for each combination of data sets included in the integration process. The results are shown in Fig. 3.9. The lowest values of average AUC ROC and AUC PR are observed when only DTI data set is used. The values increase with the inclusion of other data sets, resulting in the highest average AUC ROC when all data sets are taken into account. With all data sets taken into account by GNMTF, we use the reconstructed DTI relation matrix, $\hat{\mathbf{R}}_{23}$, to extract new drug-target interactions, as described in Sec. 3.2.2. We assess our prediction accuracy against two different large drug-target interaction databases, MATADOR [176] and CTD [175]. Out of our $225,947$ predicted DTIs, 37% have already been confirmed in MATADOR, or CTD. The list of our 20 top scoring predicted DTIs is shown in Table 3.4, out of which 17 are confirmed in CTD, or MATADOR database. Second, we investigated the literature to assess the relevance of our two top-scoring predicted

New driver	Function	Known drivers	Score	DB
ADAM32	Tumor development and inflammation	BMPR2	1.000	-
REG1P	Unknown	CLASP2	1.000	-
PCDHA2	Cell adhesion and cell-cell signaling	CHD4	1.000	-
NCR1	Natural killer cell activation	BMPR2	1.000	-
USPL1	Cell proliferation	CLASP2	1.000	-
GDPD3	Phosphoric diester hydrolase activity	DDX5	1.000	-
LECT1	Cell differentiation	CLASP2	1.000	CCGD
IL25	Cytokine activity	CDK12, CCAR1	0.975	-
BAK1	Apoptotic signaling pathway	ATRX, TFDP1, NDRG1	0.967	-
MOGAT2	Acetyltransferase activity	ATRX, TFDP1, NDRG1	0.967	-
CHAF1A	Cell cicle & DNA repair	ATRX, TFDP1, NDRG1	0.967	CCGD
PITX2	Wnt signaling pathway	ATRX, TFDP1, NDRG1	0.967	-
SIN3B	RNA polymerase II transcription corepressor activity	ATRX, TFDP1, NDRG1	0.967	-
RPL30	RNA & protein binding	ATRX, TFDP1, NDRG1	0.967	-
GRWD1	Histone methylation	ATRX, TFDP1, NDRG1	0.967	-
SNAI1	Signal transduction by p53 class mediator	ATRX, TFDP1, NDRG1	0.967	CCGD
RBMXP4	Unknown	ATRX, TFDP1, NDRG1	0.967	-
CPNE7	Transporter activity	ATRX, TFDP1, NDRG1	0.967	-
HIPK3	ATP binding & apoptotic process	ATRX, TFDP1, NDRG1	0.967	CCGD
EPOR	Signal transduction & erythropoietin receptor activity	ATRX, TFDP1, NDRG1	0.967	CCGD

Table 3.2: The list of the top scoring proposed driver genes (1^{st} column) and their associated known driver genes (2^{nd} column), with the association score (3^{rd} column), and the confirmation of their presence in CCGD database (4^{th} column).

DTIs that are not found in other databases and found evidences that they are biologically relevant. The top scoring target gene KIT (C-Kit) is particularly relevant. It is a receptor tyrosine kinase (e.g., it catalyses ATP/ADP reactions). It has been shown that unregulated activity of this gene leads to occurrence of tumors and thus, it has been proposed as a potential drug target in cancer [183]. Interestingly, we predict the drug candidate for targeting this gene to be Adenosine triphosphate (or ATP), for which a precise role in cancer is still under investigation. Increasing ATP intake is known to improve cancer patient conditions [184]. The reason could be that ATP is linked to cancer cell metabolism and either activates cell death mediated by restoration of normal mitochondrial function, or alterates the cytosolic ATP/ADP ratio, which is postulated to deactivate glycolysis (Warburg effect) in a cancer cell [185]. Another drug-target in Table 3.4 whose predicted drug is not present in CTD and MATADOR databases is GRIN3A. GRIN3A (NMDAR-l) is a sub-unit of NMDA receptor (a glutamate-regulated ion channel). NMDA receptor

New driver	Official gene name
ADAM32	ADAM metallopeptidase domain 32
REG1P	regenerating family member 1 gamma, pseudogene
PCDHA2	protocadherin alpha 2
NCR1	natural cytotoxicity triggering receptor 1
USPL1	ubiquitin specific peptidase like 1
GDPD3	glycerophosphodiester phosphodiesterase domain containing 3
LECT1	chondromodulin
IL25	interleukin 25
BAK1	BCL2 antagonist/killer 1
MOGAT2	monoacylglycerol O-acyltransferase 2
CHAF1A	chromatin assembly factor 1 subunit A
PITX2	paired like homeodomain 2
SIN3B	SIN3 transcription regulator family member B
RPL30	ribosomal protein L30
GRWD1	glutamate rich WD repeat containing 1
SNAI1	snail family transcriptional repressor 1
RBMXP4	RNA binding motif protein, X-linked pseudogene 4
CPNE7	copine 7
HIPK3	homeodomain interacting protein kinase 3
EPOR	erythropoietin receptor

Table 3.3: Official gene names of the predicted ovarian cancer driver genes.

has been proposed as a target for cancer chemotherapy [186]. It has been proposed that glutamate antagonist molecules should be used as potential drug targets [186]. Interestingly, our predicted drug, Pethidine (also known as Meperidine), is a glutamate antagonist that is known to bind NMDA receptors [187], which provides evidence that our prediction of Pethidine as a drug for targeting GRIN3A is biologically relevant. However, evidence that Pethidine can bind to GRIN3A in particular has not yet been established. Furthermore, based on the mutated genes of particular patients, we propose these newly discovered drugs (see column four in Table 3.4) for treatment of the three patient groups described in the previous section.

3.2.5 Conclusions

We propose a data fusion framework that can effectively integrate somatic mutation data along with molecular networks and drug chemical data. It is based on GNMTF method for co-clustering heterogeneous data and it can be even further extended to incorporate any number and type of data. One important advantage of our framework is that when applied to a specific cancer, it can simultaneously perform three different tasks: patient stratification into clinically different groups, novel

Gene	Drug	Score	Clusters	DB
KIT	ATP	0.873	1, 2, 3	-
GABRQ	Adinazolam	0.808	1	M
GABRQ	Fludiazepam	0.808	1	M
GABRQ	Cinolazepam	0.809	1	M
GABRQ	Clotiazepam	0.809	1	M
HTR2A	Dopamine	0.809	1, 3	C, M
GRIN3A	Pethidine	0.801	1, 2	-
CACNA2D1	Verapamil	0.761	1, 3	M
PDGFRB	ATP	0.724	1, 2	-
KDR	ATP	0.724	1, 3	C
HTR1A	Mirtazapine	0.720	1, 2	C ,M
GABRA6	Adinazolam	0.688	1	M
GABRA6	Fludiazepam	0.688	1	M
GABRA6	Cinolazepam	0.688	1	M
GABRA6	Clotiazepam	0.688	1	M
GABRA4	Adinazolam	0.687	1, 3	M
GABRA4	Fludiazepam	0.687	1, 3	M
GABRA4	Cinolazepam	0.687	1, 3	M
GABRA4	Clotiazepam	0.687	1, 3	M
CACNA1D	Magnesium Sulfate	0.676	1, 2, 3	M

Table 3.4: The list of predicted top scoring drug-target associations (first two columns), the association scores (third column), and the confirmation of their presence in CTD (C) or MATADOR (M) database (fifth column). All drugs are FDA-approved.

driver gene identification and drug repurposing predictions for treating cancer.

We apply the GNMTF-based data fusion framework to ovarian cancer patients and identify three substantially different groups of patients with different survival outcomes. In addition, from the obtained gene clusters, we identify a list of genes that we postulate as potential drivers of ovarian cancer progression due to their strong cluster associations to known ovarian cancer driver genes. We perform biomedical literature curation for the top scoring predictions, ADAM32 and REG1P, and show that they are related to cancer cell proliferation and tumor progression, while 40% of other predictions we validate in other databases. Moreover, our framework is capable of predicting new drugs that could be used for targeting mutated genes and thus, for treatment of identified groups of ovarian cancer patients. We provide a list of predicted drug-target interactions, a good number of which is matching those reported in other databases. Other, non-validated predictions for driver genes and drug-target interactions could be true, awaiting experimental validation.

Our analysis also suggests that somatic mutation data is a valuable complement to other molecular data, whose integration with those data could lead to an improvement in the performance of data fusion methods. Our approach has a po-

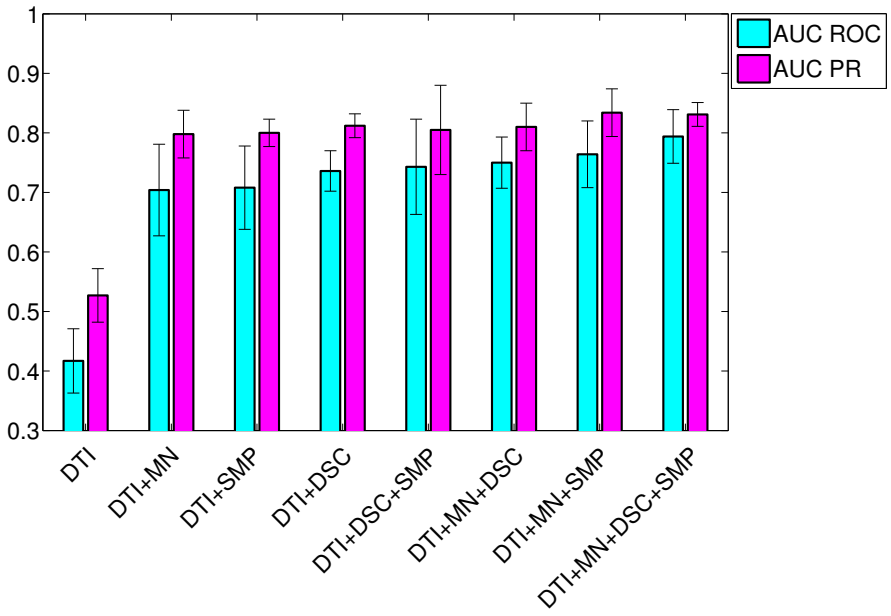


Figure 3.9: Area under the ROC and PR curves for GNMTF runs done on the combination of particular input data sets listed on x-axis. The results are sorted increasingly according to the AUC ROC values. See Fig. 3.6 for the abbreviated names of the data sets. The figure is taken from Gligorijević et al. [139].

tential to enable the derivation of new hypotheses, improve drug selection and lead to improvement in patient genomics-tailored therapeutics for cancer.

3.3 Multiple Network Alignment via network and sequence integration

Discovering patterns in networks of protein-protein interactions (PPIs) is a central problem in systems biology. Alignments between these networks aid functional understanding as they uncover important information, such as evolutionary conserved pathways, protein complexes and functional orthologs. However, the complexity of the multiple network alignment problem grows exponentially with the number of networks being aligned, and designing a multiple network aligner that is both scalable and that produces biologically relevant alignments is a challenging task that has not been fully addressed. The objective of multiple network alignment is to create clusters of nodes that are evolutionarily and functionally conserved across all networks. Unfortunately, the alignment methods proposed thus far do not meet this objective as they are guided by pairwise scores that do not utilize the entire functional and evolutionary information across all networks.

To overcome this weakness, we propose *Fuse*, a new multiple network alignment algorithm that works in two steps. First, it computes our novel protein functional similarity scores by fusing information from wiring patterns of *all* aligned PPI networks and sequence similarities between their proteins. This is in contrast with the previous tools that are all based on protein similarities in *pairs* of networks being aligned. Our comprehensive new protein similarity scores are computed by GNMTF (previusly described in Sections 3.1 and 3.2) that predicts associations between proteins whose homology (from sequences) *and* functioning similarity (from wiring patterns) are supported by *all* networks. Using the five largest and most complete PPI networks from BioGRID, we show that GNMTF predicts a large number protein pairs that are biologically consistent. Second, to identify clusters of aligned proteins over all networks, Fuse uses our novel maximum weight k-partite matching approximation algorithm. We compare Fuse with the state of the art multiple network aligners and show that: (1) by using only sequence alignment scores, Fuse already outperforms other aligners and produces a larger number of biologically consistent clusters that cover all aligned PPI networks, and (2) using both sequence alignments and topological NMTF-predicted scores leads to the best multiple network alignments thus far.

3.3.1 Introduction

Understanding the patterns in molecular interaction networks is of foremost importance in systems biology, as it is instrumental to understanding the functioning

of the cell [188]. A large number of studies focused on understanding the topology of these networks [189, 190]. Network alignment started as a pairwise problem: given two networks, aligning them means finding a node-to-node mapping (called an *alignment*) between the networks that groups together evolutionarily or functionally related proteins between the networks. These methods uncovered valuable information, such as evolutionarily conserved pathways and protein complexes [191, 192], and functional orthologs [193]. Finding these allows transfer of information across species, such as performing Herpes viral experiments in yeast or fly and then applying the insights towards understanding the mechanisms of human diseases [194].

The pairwise network alignment problem is computationally intractable due to NP-completeness of the underlying sub-graph isomorphism problem [195]. Hence, several pairwise network alignment heuristics have been proposed. Early methods, called *local network aligners*, search for small, but highly conserved sub-networks [196–198]. As such sub-networks can be duplicated, local network aligners often produce one-to-many or many-to-many mappings, in which a node from a given network can be mapped to several nodes of the other network. While these multiple mappings can indicate gene duplications, they are often biologically implausible [199]. Hence, *global network aligners*, which perform an overall comparison of the input networks and produce one-to-one mappings between the nodes of the two networks have been introduced Clark and Kalita [200].

The number of known protein-protein interactions (PPIs) increased dramatically over the last two decades thanks to the technological advances in high-throughput interaction detection techniques, such as yeast two-hybrid [201, 202] and affinity purification coupled to mass spectrometry [203]. With the availability of PPI networks of multiple species came the multiple network alignment problem, where given k networks, aligning them means to group together the proteins that are evolutionarily or functionally conserved between the networks. Similar to pairwise network alignment, multiple network alignment can be local or global, with node mappings one-to-one or many-to-many. As the complexity of the problem grows exponentially with the number of networks to be aligned, the proposed multiple network alignment algorithms use simple and scalable alignment schemes. The pioneering multiple network alignment algorithm is NetworkBLAST [204, 205], which greedily searches for highly conserved local regions in the alignment graph constructed from the pairwise protein sequence similarities. Graemlin [198] produces local multiple network alignments using a progressive alignment scheme, by successively performing pairwise alignments of the closest network pairs. IsoRank [206] and its successor IsoRankN [207] are the first multiple network aligners that do not only use pairwise sequence similarity to guide their alignment processes, but they also take into account the topology (i.e., wiring patterns) around the two nodes in their corresponding networks

to build up global many-to-many multiple network alignments, using a derivative of Google's PageRank algorithm. Smetana [208] also produces global many-to-many multiple network alignments using both pairwise sequence scores and pairwise topological scores, which are derived from a semi-Markov random walk model. While NetCoffee [209] does not use topological information to build its global one-to-one alignment, it is the first multiple network aligner in which the score for mapping two nodes does not only depend on the scores in pairs of networks, but also on their conservation across all PPI networks being aligned, by using a triplet approach similar to the multiple sequence aligner, T-Coffee [210]. Beams [211] is a fast heuristics that constructs global many-to-many multiple network alignments from the pairwise sequence similarities of the nodes by using a backbone (seed) extraction and merge strategy. CSRW [212] follows the Smetana methodology, but uses a context-sensitive random walk model. Finally, Node-Handprinting (NH) [213] produces global many-to-many multiple network alignments from topological information only, by using a progressive alignment heuristic. In the above mentioned aligners, most of the node mapping scores are local, in the sense that they only consider the sequence similarity or the topological similarity of the nodes. The only exception is NetCoffee, but its global scores are only based on sequence similarity and do not take into account the topology of the networks.

To overcome these limitations, we[4] propose Fuse, a novel multiple network alignment method that consists of two parts. In the first part, we compute novel similarity (association) scores between proteins by integrating sequence similarities and network wiring patterns over *all* proteins in *all* PPI networks being aligned. Our objective is to link together proteins whose homology (from their sequence similarity) and functioning similarity (from their wiring patterns in PPI networks) are supported by all networks. We do this by using the most general form of GNMTF technique. We apply GNMTF on the PPI networks of the five species that have the largest and the most complete sets of PPIs in BioGRID database [214]. On this dataset, the fusion process changes the values of sequence similarities between proteins based on network topologies, so that some of the sequence similarities that existed before the fusion disappear (about 40% in our experiments), while a large set of new ones is created by the *matrix completion* property of NMTF [215]. This new set of similar protein pairs is 13 times larger than the set of sequence-similar protein pairs due to fusion of sequence information with network topology. Moreover, the new set of protein pairs has a higher functional consistency (i.e., shared GO terms across the pairs) than the set of protein pairs found to be similar by sequence alignment. To control the contribution of topology versus sequence, our final *functional similarity score* for a pair of proteins is a weighted sum of their

[4]this study is a result of collaborative work with Dr. Nöel Malod-Dognin whose role was in developing and implementing novel maximum weight k-partite matching algorithm.

sequence similarity and their similarity predicted by GNMTF based on their wiring and sequence similarity with proteins in *all* networks being aligned. In the second part of Fuse, to construct a global one-to-one multiple network alignment, first we construct an edge-weighted k-partite graph, with the proteins of each of the k PPI networks being partitions of its node set and the above described functional similarity scores being edge weights. To construct a multiple network alignment, we find a maximum weight k-partite matching in this graph. As finding a maximum weight k-partite matching is NP-hard [216], and because GNMTF predicts a large number of similarities in the Fuse's k-partite graph, we propose a novel k-partite matching heuristic algorithm.

We evaluate the performance of Fuse against other state of the art multiple network aligners and show that Fuse produces the largest number of functionally consistent clusters that map proteins over all aligned networks. Unlike other aligners, Fuse is able to predict functional associations between proteins that are not sequence related (i.e., whose sequence similarity is not significant). These associations can contribute to the identification of functionally consistent clusters that cannot be identified by the previous aligners, as the previous aligners cannot predict new protein associations that are not sequence-based. Furthermore, these predicted clusters could be used for transfer of annotations across proteins of different species that are not sequence-related. Moreover, we show that Fuse is scalable and computationally more efficient than all of the previous aligners except Smetana (but Smetana's aligned proteins are not as functionally consistent as Fuse's; detailed below). Specifically, the data-fusion step is the most time consuming in Fuse with the time complexity of $O(n^3)$, where n is the total number of proteins in all PPI networks being aligned, while the alignment step has a smaller time complexity of $O(kn^2 \log n + kne)$, where k is the number of networks and e is the number of functional associations (similarity scores) between the proteins in them.

3.3.2 Methods

The PPI of each species i is represented by a graph (network), $N_i = (V_i, E_i)$, where the nodes in V_i represent proteins, and where two proteins are connected by an edge in E_i if they interact. Our multiple network alignment strategy consists of two steps. In the next two paragraphs, we give a short overview of these steps, before giving the full details of the methodology.

First, we use all PPI networks to be aligned and all the protein sequence similarities between them, as inputs into the GNMTF-based data fusion technique to compute new protein *similarity* scores between the proteins of the networks. Con-

sidering the obtained normal distribution of similarity scores for aligning the 5 PPI networks described above, we define as *significant* the scores that are in top 5%. We combine significant scores with the original sequence similarities to derive the final *functional scores* between pairs of proteins for the reasons explained in section 3.3.3. We construct an edge-weighted k-partite graph $G = (\bigcup_{i=1}^{k} V_i, E, W)$, where the node set is the union of the nodes sets (proteins) V_i of the input PPI networks; two nodes $u \in V_i$, $v \in V_j$, $i \neq j$, are connected by an edge (u, v) in E if their functional score is greater than zero; the corresponding edge weight in W is their functional score. No edge exists between nodes coming from the same subset V_i by definition of a k-partite graph.

Second, we construct a one-to-one global multiple network alignment by using an approximate maximum weight k-partite matching solver on G.

Datasets and pre-processing

From BioGRID (v3.2.111, April 25th, 2014) [214], we obtained the PPI networks of the 5 organisms having the largest and the most complete sets of physical PPIs: *Homo sapiens* (HS), *Saccharomyces cerevisiae* (SC), *Drosophila melanogaster* (DM), *Mus musculus* (MM), and *Caenorhabditis elegans* (CE). We retrieved the corresponding protein sequences from NCBI's Entrez Gene database [217] and computed their pairwise similarities using BLAST [218]. We also retrieved from NCBI's Entrez Gene database the Gene Ontology (GO) annotations of the proteins. Note that we only used experimentally validated GO annotations (i.e, excluding the annotations from computational analysis evidence such as sequence similarity) and that we additionally excluded annotations derived from protein-protein interaction experiments (code IPI). To standardize the GO annotations of proteins, similar to the evaluation methods of Singh et al. [206], Liao et al. [207], Alkan and Erten [211], we restrict the protein annotations to the fifth level of the GO directed acyclic graph by ignoring the higher-level annotations and replacing the deeper-level annotations with their ancestors at the restricted level. The network statistics are detailed in Table 3.5.

Collective GNMTF

In this case, we consider proteins belonging to different species as different data types. In the case of two species, i and j, the sequence similarity scores between their proteins are recorded in the high-dimensional relation matrix, $\mathbf{R}_{ij} \in \mathbb{R}^{n_i \times n_j}$, where, n_i is the number of proteins in the species i and n_j is the number of proteins in the species j. Entries in the relation matrix are e-values of the protein sequence

Id	# Nodes	BP Ann. (%)	MF Ann. (%)	CC Ann. (%)	# Edges
HS	14,164	37.2	23.2	9.6	127,907
SC	6,004	65.0	41.7	17.4	223,008
DM	8,125	36.1	13.4	6.3	38,892
MM	5,100	53.3	23.9	10.6	11,061
CE	3,841	35.0	7.3	4.2	7,726

Table 3.5: The five PPI networks considered in this study. For each PPI network (row), the table presents its Id (column 1), its number of nodes (column 2), its percentage of nodes that are annotated with at least one GO term from either biological process category (BP, column 3), molecular function category (MF, column 4), or cellular component (CC, column 5), and finally, its number of edges (column 6).

alignments computed by using BLAST. Specifically, we use $1 - eval$ (for $eval \leq 1$) as a measure of association between protein pairs. NMTF estimates the high-dimensional matrix, $\mathbf{R}_{ij}$ as a product of low-dimensional non-negative, latent feature matrices: $\mathbf{R}_{ij} \approx \mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T$, where, $\mathbf{G}_i \in \mathbb{R}_+^{n_i \times k_i}$ and $\mathbf{G}_j \in \mathbb{R}_+^{n_j \times k_j}$ correspond to the cluster indicator matrices of proteins in the first and the second species respectively, and $\mathbf{S}_{ij} \in \mathbb{R}^{k_i \times k_j}$ is a low-dimensional, compressed version of $\mathbf{R}_{ij}$, where the choice of rank parameters, $k_i, k_j \ll min\{n_i, n_j\}$, provides dimensionality reduction.

Some entries in the initial relation matrix $\mathbf{R}_{ij}$ are zero (due to lack of sequence similarities between the corresponding proteins) and they can be recovered from the obtained low-dimensional matrix factors using the *reconstructed relation matrix*: $\hat{\mathbf{R}}_{ij} = \mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T$ (detailed below). Here we use the *matrix completion* property to predict new and recover the existing association between proteins. To obtain the low-dimensional matrix factors, $\mathbf{G}_i, \mathbf{S}_{ij}, \mathbf{G}_j$, we solve the following optimization problem:

$$\min_{\mathbf{G}_i \geq 0, \mathbf{G}_j \geq 0} J = \min_{\mathbf{G}_i \geq 0, \mathbf{G}_j \geq 0} \parallel \mathbf{R}_{ij} - \mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j^T \parallel^2 \tag{3.13}$$

We incorporate PPI network topology as constraints into our optimization problem; violation of these constraints causes penalties to our objective function. Namely, the aim is to allow proteins interacting within a PPI network to belong to the same cluster. Interactions between proteins in PPI network, i, are represented by a graph Laplacian matrix, $\mathbf{L}_i = \mathbf{D}_i - \mathbf{A}_i$, where $\mathbf{A}_i$ is the adjacency matrix of network i and $\mathbf{D}_i$ is the diagonal degree matrix of i (i.e., diagonal entries in $\mathbf{D}_i$ are row sums of $\mathbf{A}_i$). For all five of our PPI networks we construct a Laplacian matrices, resulting in the set: $\{\mathbf{L}_1, \ldots, \mathbf{L}_5\}$.

Collective factorization of all relation matrices, $\mathbf{R}_{ij}$, $\forall i, j \in \{1, \ldots, 5\}$, and graph-regularization terms can be mathematically represented by using *block-based matrix representation*. Namely, we use a block-based representation of relation ($\mathbf{R}$)

and Laplacian ($\mathbf{L}$) matrices and low-dimensional latent feature matrices ($\mathbf{S}$ and $\mathbf{G}$) for our 5 PPI networks as follows:

$$\mathbf{R} = \begin{bmatrix} 0 & \mathbf{R}_{12} & \dots & \mathbf{R}_{15} \\ \mathbf{R}_{12}^T & 0 & \dots & \mathbf{R}_{25} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{15}^T & \mathbf{R}_{25}^T & \dots & 0 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & 0 & \dots & 0 \\ 0 & \mathbf{L}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{L}_5 \end{bmatrix};$$

$$\mathbf{S} = \begin{bmatrix} 0 & \mathbf{S}_{12} & \dots & \mathbf{S}_{15} \\ \mathbf{S}_{12}^T & 0 & \dots & \mathbf{S}_{25} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{15}^T & \mathbf{S}_{25}^T & \dots & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{G}_1 & 0 & \dots & 0 \\ 0 & \mathbf{G}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{G}_5 \end{bmatrix}$$

To simultaneously factorize all relation matrices, $\mathbf{R}_{ij} \approx \mathbf{G}_i \mathbf{S}_{ij} \mathbf{G}_j^T$, $0 \leq i, j \leq 5$, under the constraints of PPI networks, we minimize the following objective function:

$$\min_{\mathbf{G} \geq 0} J = \min_{\mathbf{G} \geq 0} \left[\parallel \mathbf{R} - \mathbf{G}\mathbf{S}\mathbf{G}^T \parallel^2 + \gamma Tr(\mathbf{G}^T \mathbf{L} \mathbf{G}) \right] \tag{3.14}$$

where, γ is a regularization parameter which balances the influence of network topologies in reconstruction of the relation matrix. The second term of equation 3.14 is the penalization term. It takes into account protein connections within the PPI network in the following way: connected pairs of proteins are represented with negative entries in the Laplacian matrix of the corresponding PPI network, and these entries act as rewards that reduce the value of the objective function, J, forcing the proteins to belong to the same cluster. Note that when $\gamma = 0$, the topology is ignored and thus Equation 3.14 is equivalent to Equation 3.13 [137].

The optimization problem (Equation 3.14) is solved by applying the algorithm following *multiplicative update rules* used to compute matrices $\mathbf{G}$ and $\mathbf{S}$ and under which the objective function, J, is non-increasing [137]. These update rules are derived by minimizing the *Langragian function*, $\mathcal{L}$, constructed from the objective function and all additional constraints, including positivity of matrix factors $\mathbf{G}$, as in paper by Wang et al. [137]. The update rule for $\mathbf{S}$ is obtained by fixing the other matrix factor, $\mathbf{G}$, and finding the roots of the equation: $\partial \mathcal{L} / \partial \mathbf{S} = 0$. A similar procedure is followed for obtaining the update rule for matrix factor $\mathbf{G}$. The multiplicative update rules, their derivation and the proof of their convergence can be found in [137].

The central idea of the collective GNMTF-based data fusion approach lies in the fact that the relation matrices are not factorized separately, but instead, are coupled by the same low-dimensional matrix factors, $\mathbf{G}_i$, which participate in their

simultaneous decomposition [219] (see Figure 3.10 for an illustration). This corresponds to the *intermediate* data fusion approach (which keeps the structure of the data while inferring a model), that has been shown to be the most accurate from all data fusion approaches [219–221].

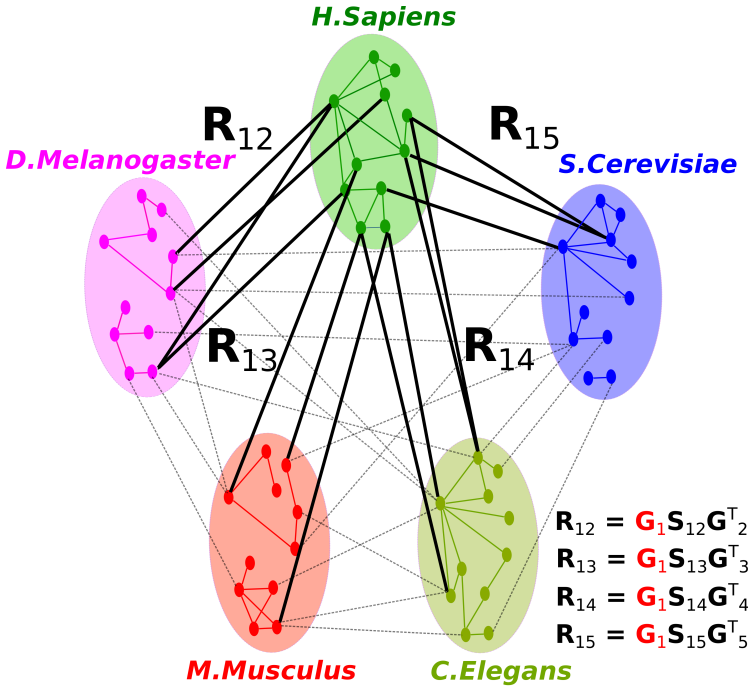


Figure 3.10: An illustration of the basic principle of GNMTF-based data fusion of 5 PPI networks. Low-dimensional matrix factor $\mathbf{G}_1$, shown in red, is shared in the decompositions of data sets represented by relation matrices: $\mathbf{R}_{12}, \mathbf{R}_{13}, \mathbf{R}_{14}, \mathbf{R}_{15}$. Therefore, the decomposition of $\mathbf{R}_{12}$ depends on the other relation matrices through the shared matrix $\mathbf{G}_1$. The figure is taken from Gligorijević et al. [140].

In our study, we use the the following values of parameters for GNMTF: (a) factorization ranks, $k_1 = 80$, $k_2 = 90$, $k_3 = 80$, $k_4 = 70$ and $k_5 = 50$, which we estimated by computing principal components of relation matrices by using Principal Component Analysis (PCA) [222]; (b) we chose the regularization parameter, $\gamma = 0.7$, since it gives the best biological quality of the predicted associations. Namely, for each value of $\gamma \in \{0., 0.001, 0.01, 0.1, 0.5, 0.7, 0.9\}$, we compute the functional consistency of the GNMTF-predicted protein similarities. We find that the highest number of highly associated and biologically consistent protein pairs is obtained when $\gamma = 0.7$, which highlights the importance of incorporating the network topology in the factorization scheme (see Section C.1 in Appendix C for details).

After the convergence of GNMTF, we compute the reconstructed relation matrices over all pairs of networks, i and j: $\hat{\mathbf{R}}_{ij} = \mathbf{G}_i\mathbf{S}_{ij}\mathbf{G}_j$. Further, we threshold the matrices by keeping only the top 5% of all associations of each protein of each species. This "thresholding strategy" leads to better biological results than sampling strategies based on statistical significance (see Appendix C Section C.3).

Since a large number of initial associations is not recovered after the GNMTF procedure (see Section 3.3.3 for details), to balance between the contribution to protein similarity from sequence and from GNMTF, we compute the final protein functional score $w(u, v)$ between proteins $u \in n_i$ and $v \in n_j$ in the k-partite network, as a linear combination of their sequence similarity $seq(u, v)$ and their GNMTF-predicted score $\hat{\mathbf{R}}_{ij}(u, v)$:

$$w(u, v) = \alpha seq(u, v) + (1 - \alpha)\hat{\mathbf{R}}_{ij}(u, v), \tag{3.15}$$

where α is a balancing parameter in $[0, 1]$ to either favour the sequence similarities (when $\alpha = 1$, only sequence similarities are used) or the novel predicted associations (when $\alpha = 0$, only GNMTF scores are used). These are the weights in the k-partite network.

Approximate maximum weight k-partite matching

Using the weighted k-partite graph representation described above, we globally align multiple networks by finding a maximum weight k-partite matching in G (defined above). The maximum weight k-partite matching problem is known to be NP-hard for $k \geq 3$ [216, 223]. Given the large number of links between the proteins across the networks produced by GNMTF, we must use a heuristic for finding an approximate solution.

To handle this large number of link and to achieve a better approximation of the maximum weight k-partite matching problem, we propose a novel maximum weight k-partite matching heuristic, which we base on the clique clustering algorithm proposed by He et al. [224] that has a low time-complexity. To this aim, we define the following graph merge operation. Let $G = (\bigcup_{i=1}^{k} V_i, E, W)$ be an edge-weighted k-partite graph, and $G[V_i, V_j]$ be the edge-weighted bi-partite subgraph of G that is induced by the two subsets of nodes V_i and V_j. Let $F_{i,j} = \{u_1 \leftrightarrow v_1, u_2 \leftrightarrow v_2, \ldots, u_l \leftrightarrow v_l\}$ be a matching of $G[V_i, V_j]$, where $u_k \leftrightarrow v_k$ means that node $u_k \in V_i$ is matched with node $v_k \in V_j$. We merge V_i with V_j into V_{ij} by identifying the mapped nodes $u_k \leftrightarrow v_k$ and by creating a corresponding *merged node* $u_k v_k \in V_{ij}$. These merged nodes inherit the edges from their parent nodes, and multiple edges are replaced by a single edge with the sum of weights of the multiple edges as the new weight of the edge. We also move into V_{ij} the nodes of V_i and V_j that are not matched. The new weighted graph G_{ij} is called the *merge* of V_j to V_i from G along $F_{i,j}$. We note that G_{ij} is an edge-weighted $(k - 1)$- partite graph.

Our approximated maximum weight k-partite matching algorithm can be seen as a progressive aligner which first maps and merges the two first networks, and then

successively adds into the "merge graph" the remaining networks (see Algorithm 2).

The performance of our algorithm depends on the order in which the networks (i.e., the partitions in the k-partite graph) are merged. We tested two different ordering strategies: merging starting from the smaller towards the larger networks, and merging networks according to the phylogenetic-tree constructed from the weights of the maximum weight bi-partite matching between the networks (i.e., merging the most similar networks first). In the rest of this section, we only report the alignments obtained by merging from the smaller network to the larger one, as it leads to the best multiple network alignment results (the comparison of the multiple network alignments that are obtained using each strategy is presented in Figure C.5 in Section C.2).

Algorithm 2 Approximate maximum weight k-partite matching.

Input $G = (\bigcup_{i=1}^{k} V_i, E, W)$
for $i = \{2, \ldots, k\}$ **do**
  Find maximum weight bipartite matching $F_{1,i}$ of $G[V_1, V_i]$
  Construct G_{1i}, the merge of V_1 and V_i from G along $F_{1,i}$
  Set $G = G_{1i}$, and relabel V_{1i} as V_1
$C = \{\emptyset\}$
for each merged node u in V_1 **do**
  Cluster C_u is the set of nodes that are merged into u
  Add C_u to C
Output C

The main operation in Algorithm 2 is finding a maximum weight matching in an induced bi-partite graph, which takes $O(n^2 \log n + ne)$ time [225, 226], when the k-partite graph has n nodes and e edges. There are $k - 1$ such operations, hence Algorithm 2 computes an approximate solution for the maximum weight k-partite matching problem in $O(kn^2 \log n + kne)$ time.

3.3.3 Results and Discussion

Biological assessment of GNMTF predicted protein similarities

The input data consist of $1,477,372$ sequence similarities between all proteins in the PPI networks of the 5 species. Using these similarities as input along with topologies of the five PPI networks, GNMTF outputs $19,175,378$ significant similarities (i.e., those obtained by keeping the top 5% of the associations of each protein that are obtained from the reconstructed relation matrices). These associations, resulting from GNMTF, cover 60% of the input sequence similarities (*reconstructed*), while the remaining associations resulting from GNMTF are *predicted*.

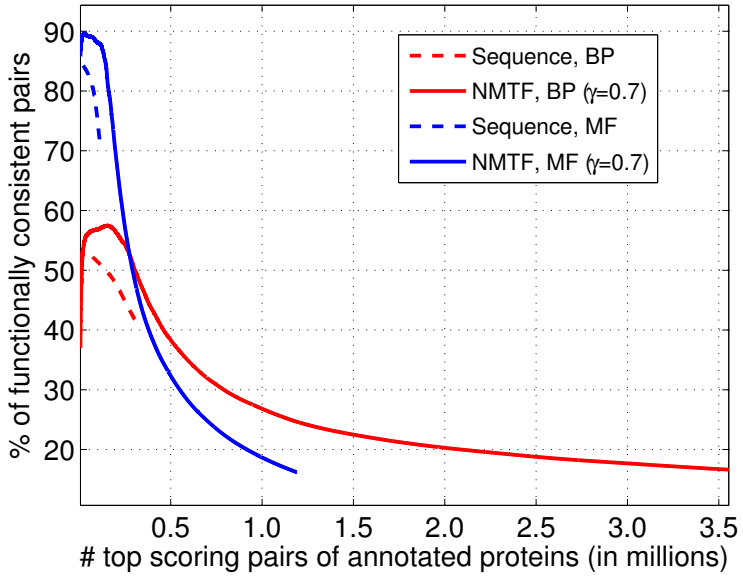


Figure 3.11: Functional consistency of GNMTF associations. For both GNMTF associations and sequence similarity of protein pairs, we plot the cumulative number of protein pairs with both proteins annotated (x-axis) against the percentages of them sharing GO terms (y-axis). Biological process (BP) and molecular function (MF) annotations are considered separately. The figure is taken from Gligorijević et al. [140].

To estimate the impact of PPI network topology on prediction of protein associations and to understand why 40% of the initial sequence similarities are not reconstructed through factorization process, we perform the following experiment: for each *reconstructed*, *predicted* and *non-reconstructed* protein pair, we count the number of sequence similarities between their neighbours in the corresponding PPI networks. For the protein pairs with reconstructed sequence similarities, we find that their neighbours share the highest number of sequence similarities, 20.4 on average. We also find that protein pairs with predicted associations share 12.1 sequence similarities between neighbouring proteins on average. In contrast, a much smaller number of sequence similar neighbours, 8.6 on average, is observed for the protein pairs with non-reconstructed similarities. This means that GNMTF induces new and reconstructs existing associations between proteins that have many sequence similar neighbours in the corresponding PPI networks. Hence, the sequence similarity of protein pairs without many sequence similar neighbours in their PPI networks will be lost in GNMTF process.

To assess the functional consistency of GNMTF's protein associations, we compute the cumulative number of associations between annotated proteins and the percentage of them sharing GO term (we considered BP and MF annotations separately). Compared with input sequence similar annotated proteins, GNMTF achieve

both larger numbers of functionally consistent paired proteins and higher functional consistency for the top-scoring pairs (Figure 3.11). This higher functional consistency is very important in the context of clustering and alignment, where the highest associations are considered first. Also, the best GNMTF scores are obtained with $\gamma = 0.7$. This means that topologies of PPI networks contribute to functional coherence of protein pairs predicted to be similar by GNMTF.

To illustrate the cases where GNMTF predicts functionally consistent proteins that cannot be identified by using only sequence similarity, we extracted from the GNMTF's predictions the pairs of proteins such that: (1) their sequence similarity is not significant (i.e., e-value ≥ 1), and (2) that share at least one level 5 molecular function GO annotation. We investigated the top scoring such pairs and fond that these new associations are relevant. E.g., the five highest scoring pairs are: CTK1 (yeast) and MAP3K7 (mouse), SGV1 (yeast) and MAP3K7 (mouse), MEK1 (yeast) and MEK2 (worm), MAP3K7 (human) and CTK1 (yeast), and SVG1 (yeast) and MAP3K7 (human). All these proteins are kinases that catalyse phosphorylation reactions.

Fusing PPI networks

We fuse the five PPI networks and assess its results against state-of-the-art multiple network aligners; Beams [211], Smetana [208], CSRW [212] and NH [213]. We tried to obtained alignments from IsorankN [207] and NetCoffee [209], but the computations did not finish after more than one week. We use BLAST e-values as input sequence scores for all methods, using $1 - evalue$ as the similarity measure. Both Fuse and Beams use parameter $\alpha \in [0, 1]$ to balance the amount of input protein sequence similarity versus network topology. For these methods, we sample α from 0 to 1, in increments of 0.1. We left the other parameters of Beams and all the parameters of Smetana, CSRW and NH at their default values.

Evaluation based on coverage. First, we compare the network alignment methods on their ability to form protein clusters that cover *all* five of the input PPI networks. We refer to these clusters as "*good*" clusters, as opposed to "*bad*" clusters that cover proteins from fewer than 5 PPI networks. The *k-coverage* is the number of clusters containing proteins from k different PPI networks. Because the number of proteins per cluster may vary, the k-coverage is also expressed in terms of the number of proteins that are in these clusters. The *total coverage* considers all clusters containing proteins from at least two networks. The coverage statistics of the alignments are summarised in Figure 3.12.

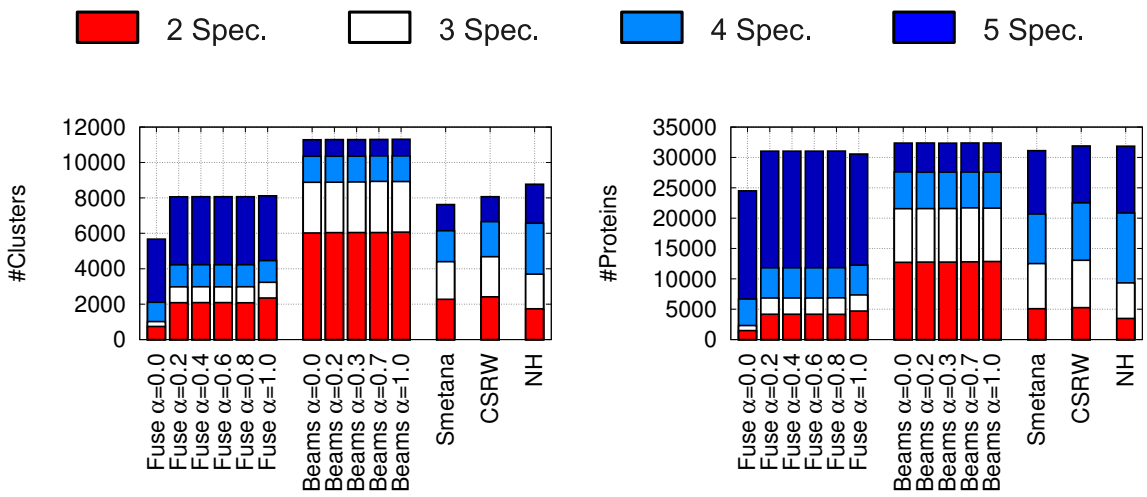


Figure 3.12: Coverage analysis. *Left:* for each alignment produced by the compared alignment methods (for a specific value of α for Fuse and Beams), the bar chart shows the number of clusters containing proteins from k species (see the colour coding on the top). *Right:* the figure shows the same, but in terms of the number of proteins in these clusters. The figure is taken from Gligorijević et al. [140].

Fuse produces a larger number of *good* clusters (i.e., containing proteins from all five species; in dark blue in Figure 3.12), producing 3,841 of such clusters. Beams achieves the highest total coverage (with up to 11,302 clusters containing proteins from two to five species), but it does so by producing the largest number of *bad* clusters (i.e., containing proteins from only two species; in red in Figure 3.12), producing up to 6,046 of such clusters, and the smallest number of good ones (937 clusters containing 4,803 proteins). When the coverage is expressed in terms of number of protein in the clusters (right panel of Figure 3.12), the total coverages of all methods are similar. However, Fuse outperforms all other methods by putting the largest number of proteins (19,205) in good clusters.

Interestingly, when Fuse uses sequence information only (i.e., when $\alpha = 1$), it already outperforms all the other approaches, which demonstrates the superiority of our alignment heuristic (see Fig. 3.12). Moreover, Fuse achieves the best coverage for $\alpha \approx 0.8$, when the functional similarity between the proteins is a combination of their sequence similarity and of their GNMTF predicted similarity, which shows the complementarity of network topology and protein sequence as sources of biological information.

Evaluation based on functional consistency. We assess functional homogeneity of the clusters obtained by each method. We say that a cluster is *annotated* if at least two of its proteins are annotated by a GO term. We say that an annotated cluster is *consistent* if all of its annotated proteins have at least one common GO

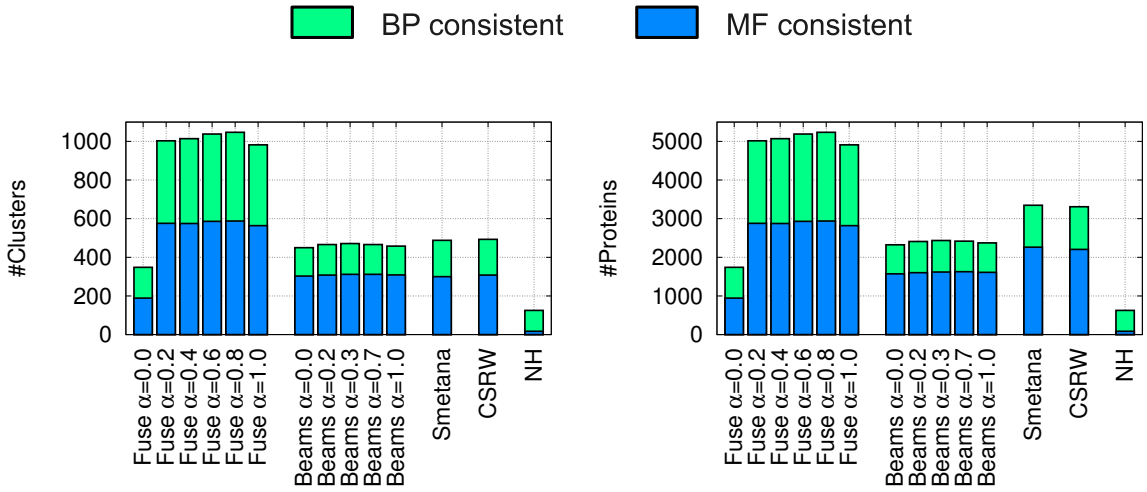


Figure 3.13: Functional consistency analysis. *Left:* for each alignment produced by the compared alignment methods (for a specific value of α for Fuse and Beams), the bar chart shows the number of clusters that contain proteins from all five species and that are BP consistent (in green) or MF consistent (in blue). *Right:* the figure shows the same, but in terms of the number of proteins in these clusters. The figure is taken from Gligorijević et al. [140].

term. The ratio of all consistent clusters to all annotated clusters we call *specificity*. Another consistency measure that is used in previous studies [207, 208, 211] is the *mean normalized entropy* (MNE). The normalized entropy of an annotated cluster c is defined as $NE(c) = -\frac{1}{\log d} \sum_{i=1}^{d} p_i \times \log p_i$, where p_i is the fraction of proteins in c with the annotation GO_i, and d represents the number of different GO annotations in c. MNE is the average of the normalized entropy of all annotated clusters. We compare Fuse, Beams, Smetana, CSRW and NH on their ability to uncover functionally conserved proteins across all input networks, by measuring the consistency, specificity and MNE of their clusters that contain proteins from all five networks (see Figure 3.13 and Table 3.6). We consider GO annotations from biological process (BP) and molecular function (MF) separately and do not consider cellular component (CC) annotations, as CC only annotate 9.7% of the proteins in the five networks.

When using sequence information only (i.e., $\alpha = 1$), Fuse already outperforms other aligners, by creating a larger number of functionally consistent clusters that consist of a larger number of proteins than previous methods (see Fig. 3.13). Fuse creates 418 clusters that are functionally consistent with respect to GO Biological Process (BP) (containing jointly 2,090 proteins), while the best competing method, Smetana, produces only 188 BP consistent clusters (containing jointly 1,086 proteins). Fuse also creates 564 functionally consistent clusters with respect to GO

		Fuse					Beams					Smetana	CSRW	NH
		α =0	0.4	0.6	0.8	1	$\alpha = 0$	0.2	0.3	0.7	1			
BP:	#C	159	439	452	**459**	418	147	158	159	154	149	188	185	108
	#P	795	2,195	2,260	**2,295**	2,090	751	809	815	790	762	1,086	1,103	540
	Spec.	6.7%	14.3%	14.8%	14.9%	14.2%	18.3%	19.2%	**19.4%**	18.8%	18.2%	14.7%	15.3%	6.3%
	MNE	**1.97**	2.05	2.05	2.05	2.07	**1.97**	2.19	2.22	2.22	**1.97**	2.16	2.06	2.04
MF:	#C	189	575	586	**588**	564	303	308	312	312	309	300	308	17
	#P	945	2,875	2,930	**2,940**	2,820	1573	1,601	1,619	1,628	1610	2,262	2,205	85
	Spec.	21.0%	40.7%	41.1%	41.3%	41.8%	66.4%	65.8%	66.0%	**68.3%**	68.2%	42.1%	45.8%	2.4%
	MNE	0.91	0.85	0.83	0.84	0.84	0.73	**0.72**	0.75	0.73	0.73	0.80	0.79	0.95

Table 3.6: Functional consistency analysis. Each column represents one of the compared alignment methods (for a specific value of α for Fuse and Beams). Numbers in cell report (from top to bottom): the number of consistent clusters (#C), the number of proteins in consistent clusters (#P), the specificity (Spec.), and the mean normalized entropy (MNE). In each row, the highest value is shown in bold.

Molecular Function (MF) (containing jointly 2,820 proteins), while the best competing method, CSRW, produces only 308 MF consistent clusters (containing jointly 2,205 proteins).

Fuse obtains the best results when it uses a combination of sequence similarities and GNMTF predicted similarities. Including predicted similarities ($\alpha = 0.8$) allows for finding up to 9% more of BP consistent clusters and for up to 4% more of MF consistent clusters than when using sequence similarity alone ($\alpha = 1$). Also, we note that these larger numbers of consistent clusters and proteins in them are not obtained at the cost of specificity, or of mean normalized entropy (see Table 3.6).

Because Fuse produces almost twice as many consistent clusters as the competing approaches, comparing methods' outputs directly may not be fair. To this end, first we score each cluster produced by Fuse with the sum of its proteins' pairwise similarity scores. Then, we sort Fuse's clusters by decreasing scores (i.e., from the cluster whose proteins are the most similar to the cluster whose proteins are the least similar) and consider the number of consistent clusters in that ordering.

To compare Fuse against other methods, but on the same number of clusters that other methods produce, we take from Fuse's output the same number of clusters as produced by other methods, that are top scoring in Fuse. E.g., Smetana outputs 1,279 BP annotated clusters out of which 188 are functionally consistent (note that Fuse outputs 3,080 BP annotated clusters out of which 459 are functionally consistent). When we take from Fuse's output the 1,279 BP highest scoring annotated clusters, we get 247 that are BP functionally consistent (in contrast to the 188 that are BP consistent by Smetana). Similarly, Fuse outperforms Smetana, CSRW and NH in both BP and MF consistent clusters (see C.3 in Section C.1) and it outperforms Beams in BP consistent clusters. For MF, Beams achieves a slightly larger number of MF consistent clusters, 312 of them out of 457 MF annotated clusters,

as opposed to Fuse's 299 clusters that are MF consistent (out of the 457 top scoring Fuse's MF annotated clusters, C.3). However, the number of proteins in the 312 of Beam's MF consistent clusters is only 1,628, while Fuse produces in total 588 MF consistent clusters that contain in total 2,940 proteins.

Another advantage of Fuse over other aligners is that in the GNMTF step, it can predict new associations between proteins whose sequence similarity is not significant (i.e., e-value ≥ 1). These associations can contribute to the identification of functionally consistent clusters that cannot be identified by other aligners. For example, Fuse can identify a cluster of 5 proteins (across 5 species) where only a subset of them have similar sequences (e-value < 1), while others are predicted to be related based on the GNMTF scores. In this case, the other aligners cannot identify 5-protein clusters because their performance is based only on sequence similarity scores, and they cannot predict new protein associations. We identify 18 clusters covering all 5 species (see Appendix C, Section C.4). One of them includes proteins: HPS6 (human), HPS6 (mouse), SEC72 (yeast), ABU-1 (worm) and NIMC2 (fly); since the first three and the last two proteins have similar sequences, if the alignment was based only on sequence similarity, it would have resulted in two clusters covering three and two species, respectively. However, because of the predicted association between HPS6 (mouse) and ABU-1 (worm), Fuse was able to identify a cluster of related proteins in all 5 species. These proteins are all involved in the immune response and also the first four proteins are located in the endoplasmic reticulum (details are in Section C.4 in Appendix C).

Fuse is also computationally efficient and scalable. The matrix factorization step is an $O(n^3)$ time operation, where n is the total number of proteins in all PPI networks. On our dataset, the matrix factorization step is the most time consuming and requires $\approx$ 10 hours to complete. The alignment step has a smaller time complexity of $O(kn^2 \log n + kne)$, where n is the number nodes in Fuse's k-partite graph (i.e., the total number of proteins in all PPI networks), and e is the total number of edges in Fuse's k-partite graph, and on our data set the alignment process requires less than 15 minutes. The time complexity of Beams is $O(nd^{k+1})$, where d is the maximum degree of a node in Beams' k-partite graph. Beams complexity becomes larger than Fuse's one when its k-partite graph becomes denser (i.e., when d tends to n). Aligning our PPI networks with Beams requires $\approx$ 78 hours. NH also has a large time complexity, $O(kn^4)$, although on our dataset it takes a short running times of $\approx$ 1 hour. Finally, Smetana and CSRW have the smallest time complexities of $O(k^3ne)$ and on our data set their computations require $\approx$ 1 hour for Smetana and $\approx$ 3 hours for CSRW.

3.3.4 Conclusions

In this section we propose Fuse, a novel global multiple network alignment algorithm which can efficiently align even the largest currently available PPI networks. Fuse computes novel similarity scores between the proteins in PPI networks by fusing all PPI network topologies and their protein sequence similarities by GNMTF. We show that these new similarities complement solely sequence-based ones: GNMTF predicts as similar 13 times more protein pairs than sequence alone does and these predicted protein pairs are functionally consistent. This demonstrates the power of data integration and contribution of network topology to sequence-based methods for finding functionally consistent proteins in different species.

We define new functional similarity scores between the proteins by combining the similarity scores obtained by GNMTF with the sequence-based ones using a user-defined balancing parameter α to either favour one or the other. Fuse uses these functional scores to construct global one-to-one multiple network alignment by using a novel maximum weight k-partite matching heuristic algorithm.

We compare the alignments of Fuse to the ones of the state-of-the art aligners, Beams, Smetana, CSRW and NH. We find that even when using solely protein sequence similarity, Fuse already outperforms all other network aligners by producing a larger number of functionally homogeneous clusters that cover all aligned networks. This shows the superiority of our k-partite matching heuristic to produce biologically meaningful multiple network alignments. Additionally, we find that the results of Fuse are even better when functional similarity scores are created from both sequence and GNMTF scores (when $\alpha = 0.8$) rather than when using sequence information only (when $\alpha = 1$). This again demonstrates complementarity of sequence and network topology in carrying biological information.

Chapter 4

Integration and analysis of homogeneous multi-relational networks

In Chapter 3, we have seen how *heterogeneous multi-relational* network representation can be useful for representation and analysis of heterogeneous biological data. We have also seen how NMTF-based methods are the most suitable for mining such networks. In this Chapter, we shift our attention to *homogeneous multi-relational* network representation and propose two novel classes of methods, based on NMF technique, for mining such networks. In particular, we focus on methods for extraction of *communities* (i.e., clusters, or modules) from such multi-relational networks. To be consistent with the majority of studies dealing with the same network representation, in the rest of this chapter, we refer to *homogeneous multi-relational* networks as *multilayer* or *multiplex* networks.

First, we start this Chapter by providing a comprehensive review of existing *state-of-the-art* methods for community detection in single-layer and multilayer networks (presented in Section 4.1). Next, in Section 4.2, we review most commonly used measures for clustering evaluation that we use in our comparison studies. Then, in Section 4.3, we introduce two novel methods, coined *Weighted Simultaneous Symmetric NMTF (WSSNMTF)* and *WSSNMTF with natural gradient (NG-WSSNMTF)*, that simultaneously decompose network layers to find a low-dimensional representation common to all layers, from which the composite communities can be extracted. Finally, in Section 4.4, we propose further improvements of these methods, and introduce a general framework, termed *Network Fusion for Composite Community Extraction (NF-CCE)*, composed of four novel methods based on four different NMF techniques that are extended for collective decompo-

sition of network layers. We conduct extensive experiments on variety of different types of networks, including biological, social, telecommunications, citation and brain networks, and demonstrate a superior clustering performance of our methods over the existing *state-of-the-art* methods. Results of this chapter have previously been published in Gligorijević et al. [227] or prepared for publication in Gligorijević et al. [228].

4.1 Introduction: from single-layer to multilayer networks

Networks along with their theoretical foundations are powerful mathematical tools for representing, modeling, and analyzing complex systems arising not only in biology, but also in several other scientific disciplines including sociology, physics, and engineering among others [229]. Concretely, social networks, economic networks, biological networks, telecommunications networks, etc. are just a few examples of graphs in which a large set of entities (or agents) correspond to *nodes* (or *vertices*) and relationships or interactions between entities correspond to *edges* (or *links*). Structural analysis of these networks have yielded important findings in the corresponding fields [230, 231].

Community detection (also known as *graph clustering* or *module detection*) is one of the foremost problems in network analysis. It aims to find groups of nodes (i.e., *clusters, modules* or *communities*) that are more densely connected to each other than they are to the rest of the network [232]. Even thought the volume of research on community detection is large, e.g., [233–241], the majority of these methods focus on networks with only one type of relations between nodes (i.e., networks of single type interaction). However, many real-world systems are naturally represented with multiple types of relationships, or with relationships that chance in time. Such systems include subsystems or layers of connectivity representing different modes of complexity (see also Section 2.1 for a brief introduction into these types of networks, where we refer to these networks as *homogeneous multi-relational* networks). For instance, in social systems, users in social networks engage in different types of interactions (e.g., personal, professional, social, etc.). In biology, different experiments or measurements can provide different types of interactions between genes. Reducing these networks to a single type interactions by disregarding their multiple modalities is often a very crude approximation that fails to capture a rich and holistic complexity of the system. In order to encompass a multimodal nature of these relations, a *multilayer network* representation has been proposed [46]. Multilayer networks (also known as *multidimensional*, *multiview* or *multiplex*

networks) have recently attracted a lot of attention in network science community. They can be represented as a set of graph layers that share a common set of nodes, but different set of edges in each layer (see Fig. 4.1 for an illustration). With the emergence of this network representation, finding composite communities across different layers of multilayer network has become a new challenge [46, 242].

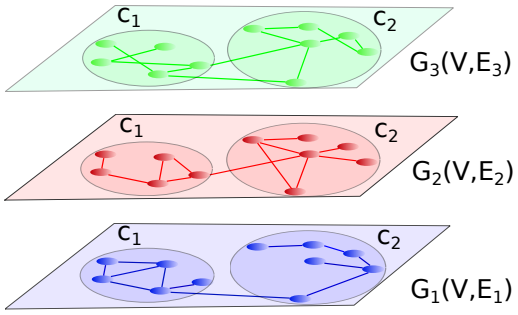


Figure 4.1: An example of multilayer networks with 11 nodes present in three complementary layers denoted in different colors. Two different communities across all three layers can be identified. The figure is taken from Gligorijević et al. [228].

Before introducing our methods, here we provide a systematic review of existing methods for single-layer and multilayer networks. To make this chapter self-contained, we start our review by repeating basic network definitions and concepts previously introduced in Section 2.1.

4.1.1 Single-layer (monoplex) networks

In graph theory, a monoplex network (graph) can be represented as an ordered pair, $G = (V, E)$, where V is a set of $n = |V|$ vertices or nodes, and E is a set of $m = |E|$ edges or links between the vertices [243]. An adjacency matrix representing a graph G is denoted by $\mathbf{A} \in \{0, 1\}^{n \times n}$, where $\mathbf{A}_{ij} = 1$ if there is an edge between vertices i and j, and $\mathbf{A}_{ij} = 0$ otherwise. Most of the real-world networks that we consider here in this chpater are represented as *edge-weighted graphs*, $G = (V, E, w)$, where $w : E \to \mathbb{R}$ assigns real values to edges. In this case, the adjacency matrix instead of being a binary matrix, is a real one i.e., $\mathbf{A} \in \mathbb{R}^{n \times n}$, with entries characterizing the strength of association or interaction between the network nodes.

Although, there is no universally accepted mathematical definition of the community notion in graphs, the probably most commonly accepted definition is the following: a community is a set of nodes in a network that are connected more densely among each other than they are to the rest of the network [46]. Hence, the problem of community detection is as follows: given an adjacency matrix $\mathbf{A}$ of one network with n nodes and k communities, find the community assignment

of all nodes, denoted by matrix $\mathbf{H} \in \{0,1\}^{n \times k}$, where $\mathbf{H}_{ij} = 1$ if nodes i belongs to community j, and $\mathbf{H}_{ij} = 0$ otherwise. We consider the case of *non-overlapping* communities, where a node can belong to only one community, i.e., $\sum_{j=1}^{k} \mathbf{H}_{ij} = 1$.

To address the community detection problem in monoplex networks, several methods have been proposed. Comprehensive surveys of these methods are [232] and [241]. Here, we briefly review some of the most representative approaches, including *graph partitioning, spectral clustering, hierarchical clustering, modularity maximization, statistical inference and structure-based methods*, as well as method that rely on *non-negative matrix factorizations*:

- *Graph partitioning* aims to group nodes into partitions such that the *cut size*, i.e., the total number of edges between any two partitions, is minimal. Two widely used graph partitioning algorithms that also take into account the size of partitions are *Ratio Cut* and *Normalized Cut* [244]. Graph partitioning algorithms can be alternatively defined as spectral algorithms in which the objective is to partition the nodes into communities based on their eigenvectors obtained from *eigendecomposition of graph Laplacian matrix* [245].

- In *hierarchical clustering* the goal is to reveal network communities and their hierarchical structure based on a similarity (usually topological) measure computed between pairs of nodes [246].

- *Modularity-based* algorithms are among the most popular ones. Modularity was designed to measure the strength of partition of a network into communities. It is defined as a fraction of the edges that fall within the community minus the expected fraction when these edges are randomly distributed [233, 247]. Various algorithms have been proposed for modularity optimization, including greedy techniques, simulated annealing, spectral optimization, etc. [232].

- *Statistical inference* methods aims at fitting the *generative model* to the network data based on some hypothesis. Most commonly used statistical inference method for community detection is the *stochastic block model*, that aims to approximate a given adjacency matrix by a block structure [248]. Each block in the model represents a community.

- *Structure-based* methods aim to find subgraphs representing meta definitions of communities. Their objective is to find maximal cliques, i.e., the cliques which are not the subgraph of any other clique. The union of these cliques form a subgraph, whose components are interpreted as communities [249].

- More recently, graph clustering methods that rely on the *Non-Negative Matrix Factorization* (NMF) [250] have been proposed e.g., [251, 252]. Their goal is to approximate a symmetric adjacency matrix of a given network by a product of two non-negative, low-rank matrices, such that they have clustering interpretation, i.e., they can be used for assigning nodes to communities. The proposed methods here, follow this line of research but as opposed to the existing methods [251, 252], they can effectively handle multilayer networks.

4.1.2 Multilayer networks

A multilayer network is a set of N monoplex networks (or layers), $G_i(V, E_i)$, for $i = 1, \ldots, N$. The number of nodes in each layer is the same, $n = |V|$, while the connectivity pattern and the distribution of links in each layer differs, $m_i = |E_i|$ (see Fig. 4.1). Similarly to monoplex networks, we consider the case where each layer represents a weighted, undirected graph, i.e., $G_i(V, E_i, w_i)$. A multilayer network can be represented as a set of adjacency matrices encoding connectivity patterns of individual layers, $\mathbf{A}^{(i)} \in \mathbb{R}^{n \times n}$, for $i = 1, \ldots, N$. The goal of community detection in multilayer networks is to infer shared, latent community assignment that best fits all given layers. Given that each layer contains incomplete and complementary information, this process of finding shared communities by integrating information from all layers is also known in the literature as *network integration (fusion)* [253, 254].

Unlike the case of monoplex networks, research on community detection in multilayer networks is scarce. Existing methods extract communities from multilayer networks first by aggregating the links of all layers into a single layer, and then applying a single-layer method to that single layer [254–256]. However, this approach does not account for shared information between layers and treats the noise present in each layer uniformly. Clearly, this is not the case in real-world multilayer networks where each level is contaminated by different noise in terms of magnitude and, possibly, distribution. Thus, by aggregating links from different layers the noise in the aggregated layer significantly increases, resulting in a poor community detection performance.

Current state-of-the-art methods are built on monoplex approaches and further generalized to multilayer networks. They can be divided into the following categories:

- *Modularity-based* approaches generalize the notion of modularity from single-layer to multilayer networks. Namely, to alleviate the above mentioned lim-

itations, the Principal Modularity Maximization (PMM) [254] has been proposed. First, for each layer, PMM extracts structural features by optimizing its modularity, and thus significantly denoising each layer; then, it applies PCA on concatenated matrix of structural feature matrices, to find the principal vectors, followed by K-means to perform clustering assignment. The main drawback of this approach is that it treats structural feature matrices of all layers on equal basis (i.e., it is not capable of distinguishing between more and less informative network layers, or complementary layers). Even though the noise is properly handled by this method, the *complementarity aspect* cannot be captured well by the integration step.

- *Spectral clustering* approaches that generalize the eigendecomposition from single to multiple Laplacian matrices representing network layers. One of the *state-of-the-art* spectral clustering methods for multilayer graphs is the Spectral Clustering on Multi-Layer (SC-ML) [257]. First, for each network layer, SC-ML computes a subspace spanned by the principal eigenvectors of its Laplacian matrix. Then, by interpreting each subspace as a point on Grassmann manifold, SC-ML merges subspaces into a consensus subspace from which the composite clusters are extracted. The biggest drawback of this methods is the underlying spectral clustering that always finds tight and small-scale and, in some cases, almost trivial communities. For example, SC-ML cannot adequately handle network layers with missing or weak connections, or layers that have disconnected parts.

- *Information diffusion-based* approaches that utilize the concept of diffusion on networks to integrate network layers. One of such methods is Similarity Network Fusion (SNF) proposed by Wang *et al.* [258]. SNF captures both shared and complementary information in network layers. It computes a fused matrix from the similarity matrices derived from all layers through parallel interchanging diffusion process on network layers. Then, by applying a spectral clustering method on the fused matrix they extract communities. However, for sparse networks, the diffusion process, i.e., information propagation, is not very efficient and it may results in poor clustering performance.

- *Matrix and tensor factorization-based* approaches that utilize collective factorization of adjacency matrices representing network layers. A few matrix and tensor decomposition-based approaches have been proposed so far [259–262]. Tang *et al.* [259] introduced the Linked Matrix Factorization (LMF) which fuses information from multiple network layers by factorizing each adjacency matrix into a layer-specific factor and a factor that is common to all network layers. Dong *et al.* [260], introduced the Spectral Clustering with Generalized Eigendecomposition (SC-GED) which factorizes Laplacian matrices in-

stead of adjacency matrices. Papalexakis *et al.* [261] proposed GraphFuse, a method for clustering multilayer networks based on sparse PARAllel FACtor (PARAFAC) decomposition [263] with non-negativity constraints. Cheng *et al.* introduced Co-regularized Graph Clustering based on NMF (CGC-NMF). They factorize each adjacency matrix using symmetric NMF while keeping the Euclidean distance between their non-negative low-dimensional representations small. One of the major limitations of all of these factorization methods is that they treat each network layer on an equal basis and, unlike PMM or SC-ML, for example, they cannot filter out irrelevant information or noise.

To alleviate the drawbacks of the aforementioned methods in Sections 4.3 and 4.4 we introduce two novel classes of methods that are built upon NMF decomposition methods.

4.2 Clustering evaluation measures

Here we discuss the evaluation measures used in our experiments to evaluate and compare the performance of our proposed methods with the above described *state-of-the-art* methods. Given that we test our methods on multilayer networks with known and unknown ground truth cluster, we distinguish between two sets of measures:

- *Known ground truth.* We use the following three widely used clustering accuracy measures: *Purity* [264], *Normalized Mutual Information (NMI)* [265] and *Adjusted Rand Index (ARI)* [265]. All three measures provide a quantitative way to compare the computed clusters $\Omega = \{\omega_1, \ldots, \omega_k\}$ with respect to the ground truth classes: $C = \{c_1, \ldots, c_k\}$. *Purity* represents percentage of the total number of nodes classified correctly, and it is defined as [264]:

$$Purity(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j|$$

  where N is the total number of objects, and $|\omega_k \cap c_j|$ represents the number of nodes in the intersection of ω_k and c_j. To trade-off the quality of the clustering against the number of clusters we use *NMI*. NMI is defined as [265]:

$$NMI(\Omega, C) = \frac{I(\Omega; C)}{|H(\Omega) + H(C)|/2}$$

  where I is the mutual information between node clusters Ω and classes C, while $H(\Omega)$ and $H(C)$ represent the entropy of clusters and classes respectively. Finally, *Rand Index* represents percentage of true positive (TP) and

true negative (TN) decisions assigns that are correct (i.e., accuracy). It is defined as:

$$RI(\Omega, C) = \frac{TP + TN}{TP + FP + FN + TN}$$

where, FP and FN represent false positive and false negative decisions respectively. ARI is defined to be scaled in range $[0, 1]$ [265]. All three measures are in the range $[0, 1]$, and the higher their value, the better clustering quality is.

- *Unknown ground truth.* For biological networks, the ground truth clusters are unknown and evaluating the clustering results becomes more challenging. In order to evaluate the functional modules identified by our methods, we use Gene Ontology (GO) [21], a commonly used gene annotation database. GO represents a systematic classification of all known protein functions organized as well-defined terms (also known as GO terms) divided into three main categories, namely Molecular Function (MF), Biological Process (BP) and Cellular Component (CC) [21]. GO terms (i.e., annotations), representing gene functions, are hierarchically structured where low-level (general) terms annotate more proteins than high-level (specific) terms. Thus, for each gene in a network, we create a list of its corresponding BP GO term annotations. We then analyze the consistency of each cluster, i, obtain by our method, by computing the *redundancy* [234], R_i as follows:

$$R_i = 1 - \frac{\Big(- \sum_{l=1}^{N_{GO}} p_l \log_2 p_l \Big)}{\log_2 N_{GO}}$$

where, N_{GO} represents the total number of GO terms considered and p_l represents the relative frequency of GO term l in cluster i. Redundancy is based on normalized Shannon's entropy and its values range between 0 and 1. For clusters in which the majority of genes share the same GO terms (annotations) redundancy is close to 1, whereas for clusters in which the majority of genes have disparate GO terms the redundancy is close to 0. When comparing clustering results obtained by different methods, we use the value of redundancy averaged over all clusters. Furthermore, the redundancy is a very suitable measure for clustering performance comparisons because its value does not depend on the number of clusters and unlike others evaluation measures for biological network clustering [266], it is parameter-free.

  In addition to redundancy, we also perform *GO term enrichment analysis* [267], on each individual gene cluster, to further examine their functional ho-

mogeneity. Using GO annotations of genes in each individual cluster, enrichment analysis finds GO terms that are over-represented in that cluster. To do so, for each GO term, GO_j, we compute the p-value, p_j, that represents the probability that a given cluster of genes is enriched in GO_j term purely by chance. The p-value is computed using cumulative hypergeometric cumulative distribution as follows:

$$p_j = 1 - \sum_{i=0}^{X-1} \frac{\binom{|GO_j|}{i}\binom{n-|GO_j|}{|C|-i}}{\binom{n}{|C|}}$$

where, n is the number of genes in the network, $|GO_j|$ is the total number of genes annotated with GO_j, X is the number of genes in cluster C annotated with GO_j and $|C|$ is the total number of genes in cluster C; p_j represents the probability that at least X genes out of total $|C|$ genes in cluster C are annotated with GO_j, given the proportion of genes in the network annotated with GO_j. The smaller value of p_j indicates the larger significance association between GO term GO_j and the group of genes in cluster C. After computing p-values for each cluster obtained by our methods, we apply *Benjamini-Hochberg correction procedure* [268] to decrease the false discovery rate (FDR) and a probability of obtaining significant p-values in a larger number of experiments merely by chance. Thus, in all our experiments, we report the *adjusted* p-values computed by applying this procedure. Also, we use $p_j \leq 0.05$ as a threshold, below which we consider a cluster to be statistically significantly enriched in GO_j.

4.3 Decomposition of network layers into a common non-negative representation

Here, we propose two novel methods for network fusion and community extraction from multiple network layers. Both methods are based on Symmetric Non-negative Matrix Tri-Factorization (*SNMTF*) [122, 251], in which an adjacency matrix, representing a graph, is decomposed into a product of three low-dimensional non-negative latent feature matrices. We extend this approach on multiple graph layers, and we introduce our methods *Weighted Simultaneous SNMTF (WSSNMTF)* and *WSSNMTF with Natural Gradient (NG-WSSNMTF)*. Both methods simultaneously factorize adjacency matrices representing network layers by using SNMTF, while keeping the low-dimensional latent feature matrices shared across the decomposition; this enables the structural information shared between graph layers to be taken into account during decomposition. Furthermore, both methods take into account the

missing values in each matrix (i.e., the missing links in each network layer). To this end, for each layer, we introduce a *symmetric weight matrix*, in which the entries are equal to one for the observed and zero for missing edges. Thus, we formulate the objective function to factorize only observed values. Furthermore, we impose sparsity penalties to our learned low-dimensional latent feature matrices by adding ℓ_1 norm regularization terms to our objective function. This will make the learned matrices more robust to noise and outliers, which is the case for many incomplete graph layers. Consequently, this leads to much stable clustering results. To the best of our knowledge, these are the first methods for multilayer community detection that consider missing and incomplete network layers. While *WSSNMTF* uses regular gradient, *NG-WSSNMTF* uses normal gradient to optimize the objective function by imposing the low-dimensional latent feature matrices to be orthonormal, and thus, leading to better clustering results (see Section 4.3.4).

4.3.1 Weighted Simultaneous SNMTF: formulation of the problem

In a multilayer network, each $\mathbf{A}^{(i)}$ is constructed over the same number of nodes n. For some network layers, this may result in $\mathbf{A}^{(i)}$ having zero rows and columns. Therefore, to treat this as missing values properly, for each network layer we define a *weight matrix*, $\mathbf{W}^{(i)}$, introduces in the following way:

$$\forall i, \quad \mathbf{W}^{(i)}_{ab} = \begin{cases} 1 & \text{if } a \text{ is connected with } b \text{ in layer i,} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

In analogy with the SNMTF, we tri-factorize each symmetric matrix $\mathbf{A}^{(i)}$ into a product of three non-negative matrices in the following way:

$$\mathbf{A}^{(i)} \approx \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T, \quad \text{where, } \mathbf{S}^{(i)} \geq 0, \quad \mathbf{H} \geq 0$$

Note that matrix $\mathbf{H}$, also called *soft cluster indicator matrix*, is shared across the decompositions. To factorize only the observed entries in matrices (i.e., the observed links) we formulate the following objective function:

$$\mathcal{L} = \sum_{i=1}^{N} \parallel \mathbf{W}^{(i)} \circ (\mathbf{A}^{(i)} - \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T) \parallel^2 + \sum_{i=1}^{N} \eta_i \parallel \mathbf{S}^{(i)} \parallel_1 \tag{4.2}$$

where $\circ$ denotes Hadamard product of matrices; ℓ_1 norm of the second terms imposes

sparsity constraint to $\mathbf{S}^{(i)}$ matrices. Parameters η_i for $i \in \{1, \ldots, N\}$ are trade-off parameters for balancing the influence between the two terms.

4.3.2 Optimization with ordinary gradient

We minimize the objective function shown in (4.2) under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{S}^{(i)} \geq 0$ for $\forall i \in \{1, \ldots, N\}$. We derive the update rules for minimizing the objective function following the procedure from the constrained optimization theory [130]. Since the objective function is not jointly convex in both $\mathbf{H}$ and $\mathbf{S}^{(i)}$, we present an alternating procedure to find a local optimal solution. That is, we optimize the objective function with respect to one variable while fixing others. The procedure is repeated until convergence.

We follow the strategy employed in the derivation of NMF [250] to obtain the multiplicative update rules for both $\mathbf{S}^{(i)}$ and $\mathbf{H}$ matrices that can be used for finding a local minimum of the optimization problem (4.2).

The derivative of the objective function with respect to $\mathbf{S}^{(i)}$ is as follows[1]:

$$\begin{aligned}\nabla_{\mathbf{S}^{(i)}}\mathcal{L} &= -\overbrace{2\mathbf{H}^T(\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)})\mathbf{H}}^{[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^-} + \overbrace{2\mathbf{H}^T(\mathbf{W}^{(i)} \circ \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T)\mathbf{H} + 2\frac{1}{2}\eta_i}^{[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^+} \\ &= [\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^+ - [\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^-\end{aligned}$$

where, $[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^+ \geq 0$, $[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^- \geq 0$ are non-negative terms.

Using the Karush-Kuhn-Tucker (KKT) complementarity condition [130] for the non-negative constraint on $\mathbf{S}^{(i)}$, $\mathbf{S}^{(i)} \leftarrow \mathbf{S}^{(i)} \circ \left(\frac{[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^-}{[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^+}\right)^{\frac{1}{2}}$, we obtain the following multiplicative update rule for each $\mathbf{S}^{(i)}$ matrix:

$$\mathbf{S}^{(i)} \leftarrow \mathbf{S}^{(i)} \circ \sqrt{\frac{\mathbf{H}^T(\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)})\mathbf{H}}{\mathbf{H}^T(\mathbf{W}^{(i)} \circ \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T)\mathbf{H} + \frac{1}{2}\eta_i}} \tag{4.3}$$

The derivative of the objective function with respect to $\mathbf{H}$ is as follows:

[1]In the derivation procedure we use the following mathematical properties: $Tr(\mathbf{ABC}) = Tr(\mathbf{BCA}) = Tr(\mathbf{CAB})$; $Tr(\mathbf{A}^T) = \mathrm{Tr}(\mathbf{A})$ and $\| \mathbf{A} \|_F^2 = Tr(\mathbf{A}\mathbf{A}^T)$; if $\mathbf{A} \geq 0$ then $\| \mathbf{A} \|_1 = Tr(\mathbb{1}^T\mathbf{A})$;$Tr((\mathbf{W}^T \circ \mathbf{A}^T)(\mathbf{W} \circ \mathbf{B})) = Tr(\mathbf{A}^T(\mathbf{W} \circ \mathbf{W} \circ \mathbf{B}))$

$$\nabla_{\mathbf{H}}\mathcal{L} = -\overbrace{\sum_{i=1}^{N} 4(\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)})\mathbf{H}\mathbf{S}^{(i)}}^{[\nabla_{\mathbf{H}}\mathcal{L}]^-} + \overbrace{\sum_{i=1}^{N} 4(\mathbf{W}^{(i)} \circ \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T)\mathbf{H}\mathbf{S}^{(i)}}^{[\nabla_{\mathbf{H}}\mathcal{L}]^+}$$

Again, from the Karush-Kuhn-Tucker (KKT) complementarity condition [130] for the non-negative constraint on $\mathbf{H}$, $\mathbf{H} \leftarrow \mathbf{H} \circ \left(\frac{[\nabla_{\mathbf{H}}\mathcal{L}]^-}{[\nabla_{\mathbf{H}}\mathcal{L}]^+}\right)^{\frac{1}{4}}$, we obtain the following update rule for $\mathbf{H}$ matrix:

$$\mathbf{H} \leftarrow \mathbf{H} \circ \left(\frac{\sum_{i=1}^{N}(\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)})\mathbf{H}\mathbf{S}^{(\mathbf{i})}}{\sum_{i=1}^{N}(\mathbf{W}^{(i)} \circ \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T)\mathbf{H}\mathbf{S}^{(\mathbf{i})}}\right)^{\frac{1}{4}} \tag{4.4}$$

Given randomly initialized non-negative matrices $\mathbf{H}$ and $\mathbf{S}^{(i)}$, we use multiplicative update rules (4.3) and (4.4) to find a local minimum of the optimization problem (4.2).

4.3.3 Optimization with natural gradient

Ding et al. [122] have proved that the orthonormality constraint imposed on non-negative low-dimensional latent feature matrix, $\mathbf{H}$, in SNMTF leads to a more rigorous clustering interpretation. Moreover, under this constraint, they showed the equivalence between NMF and k-means clustering [122]. We adopt the same idea in our optimization problem (4.2) by imposing the orthonormality constraint on the low-dimensional non-negative latent feature matrix $\mathbf{H}$: $\mathbf{H}^T\mathbf{H} = \mathbf{I}$. On the other hand, it can be shown that columns of $\mathbf{H}$ matrix span a vector subspace known as Grassmann manifold $\mathbb{G}(k, n)$, i.e., $span(\mathbf{H}) \in \mathbb{G}(k, n)$ [269]. Moreover, Amari in [270] has showed that when an optimization problem is defined over a Grassmann manifold, the ordinary gradient of the optimization function does not represent its steepest direction, but *natural gradient* does.

Therefore, we define a natural gradient to optimize our objective function (4.2) under the orthornormality constraint. Following Panagakis et al. [269], we introduce a tangent vector, $\mathbf{\Delta} \in \mathbb{R}^{n \times k}$ at $\mathbf{H}$ that satisfies: $\mathbf{H}^T\mathbf{\Delta} = 0$. The projection onto the tangent space can be given as: $\mathbf{\Pi}_{\mathbf{H}} = \mathbf{I} - \mathbf{H}\mathbf{H}^T$. By fixing $\mathbf{S}_i$ and using the projection, we can define a natural gradient of $\mathcal{L}(\mathbf{H})$ on Grassmann manifold at $\mathbf{H}$ in the following way:

$$\widetilde{\nabla}_{\mathbf{H}}\mathcal{L} = \mathbf{\Pi}_{\mathbf{H}}\nabla_{\mathbf{H}}\mathcal{L} = \nabla_{\mathbf{H}}\mathcal{L} - \mathbf{H}\mathbf{H}^T\nabla_{\mathbf{H}}\mathcal{L}$$

where, $\nabla_{\mathbf{H}}\mathcal{L}$ is the ordinary gradient, computed as in 4.3.2.

Following the Karush-Kuhn-Tucker (KKT) complementarity condition [130] and preserving the non-negativity of $\mathbf{H}$, we re-write the update rule for $\mathbf{H}$ matrix using the natural gradient: $\mathbf{H} \leftarrow \mathbf{H} \circ \left(\frac{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{L}]^-}{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{L}]^+}\right)^{\frac{1}{4}}$, where $\widetilde{\nabla}_{\mathbf{H}}\mathcal{L}$ can be written in the following way:

$$\widetilde{\nabla}_{\mathbf{H}}\mathcal{L} = [\nabla_{\mathbf{H}}\mathcal{L}]^+ - [\nabla_{\mathbf{H}}\mathcal{L}]^- - \mathbf{H}\mathbf{H}^T[\nabla_{\mathbf{H}}\mathcal{L}]^+ + \mathbf{H}\mathbf{H}^T[\nabla_{\mathbf{H}}\mathcal{L}]^-$$

The multiplicative update rule for $\mathbf{H}$ is:

$$\mathbf{H} \leftarrow \mathbf{H} \circ \left(\frac{[\nabla_{\mathbf{H}}\mathcal{L}]^- + \mathbf{H}\mathbf{H}^T[\nabla_{\mathbf{H}}\mathcal{L}]^+}{[\nabla_{\mathbf{H}}\mathcal{L}]^+ + \mathbf{H}\mathbf{H}^T[\nabla_{\mathbf{H}}\mathcal{L}]^-}\right)^{\frac{1}{4}} \tag{4.5}$$

The update rule for $\mathbf{S}$ stays the same as in (4.3).

4.3.4 Experiments

In this section, we evaluate the performance of our proposed algorithms *WSSNMTF* and *NG-WSSNMTF* on several synthetic and real world multilayer networks. First, we describe the multilayer networks that we use for clustering evaluation, and then provide a brief description of algorithms that we use for clustering comparison. Finally, we present the results in terms of the three different clustering evaluation measures introduced in Section 4.2.

Synthetic and real-world datasets

Synthetic data generation. We generate three synthetic multilayer graphs each with $N = 3$ different layers (see Fig. 4.2) by using the planted partition model approach described in [261]. For each multilayer network, we choose $n = 400$ nodes partitioned into three communities with $n_1 = 150$, $n_2 = 50$ and $n_3 = 200$ nodes respectively in each community. For each layer, we split the corresponding adjacency matrix into blocks defined by the partition. Entries in each diagonal block, are filled with ones randomly, with probability p_{ii}, representing the edge connectivity

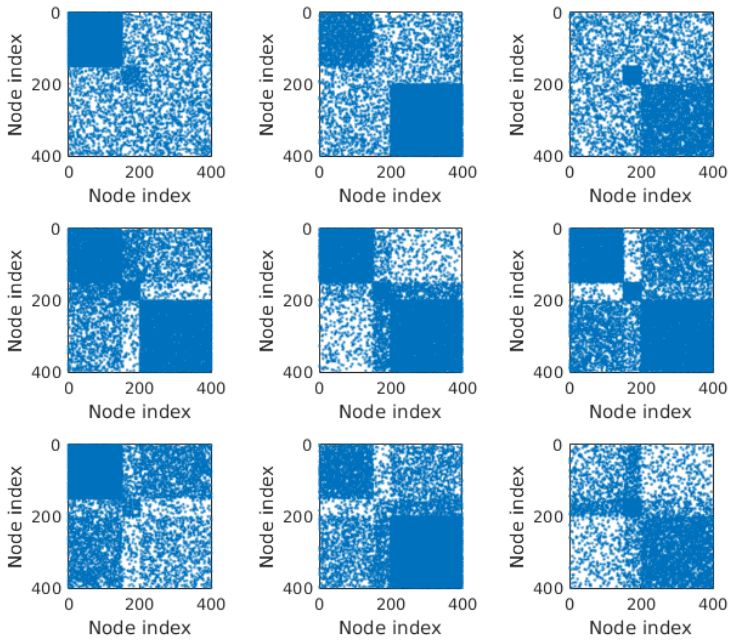


Figure 4.2: Spy-plots of three random synthetic multilayer networks with three layers. (top) SYNT-1 is composed of three complementary layers with communities of different connectivity probabilities and fixed probability of noise between communities. (middle) SYNT-2 is composed of three layers with different levels of noise between communities and fixed connectivity probability of communities. (bottom) SYNT-MIX is composed of three layers generated by combining approaches from SYNT-1 and SYNT-2. The figure is taken from Gligorijević et al. [227].

probability; it is also referred as community's edge density. We also add random noise between each pair of blocks, ij, with probability p_{ij}, representing links between communities. The larger the values of p_{ij} are the harder the clustering is. Similarly, the smaller the values of p_{ii} are the harder the clustering is.

To demonstrate the ability of *WSSNMTF* and *NG-WSSNMTF* to efficiently integrate *complementary* network layers and to demonstrate their *robustness* against different levels of noise we generate three different synthetic random multilayer networks. *SYNT-1* (Fig. 4.2, top) has three layers with different community connectivity and fixed noise between communities. Connectivity probabilities of communities in each layer correspond to one cyclic permutation of the following probabilities: $p_{11} = 0.4$, $p_{22} = 0.2$, $p_{33} = 0.1$. Noise between communities is $p_{12} = p_{13} = p_{23} = 0.05$ and it is the same in all layers. In that way, in each layer we have only one community that is fully represented, while other two are partially represented. *SYNT-2* (Fig. 4.2, middle) has fixed probabilities of communities' connectivities (i.e., $p_{11} = 0.4$, $p_{22} = 0.2$, $p_{33} = 0.1$) but noise between blocks in each layer corresponds to one cyclic permutation of the following probabilities: $p_{12} = 0.05$, $p_{13} = 0.03$,

$p_{23} = 0.01$. *SYNT-MIX* corresponds to the mixture of *SYNT-1* and *SYNT-2* and it is generated by permuting both p_{ii} and p_{ij} probabilities when generating different layers.

Real data description We use three different real data multilayer networks. Two of them are constructed from bibliographic datasets, and the third one is a mobile phone dataset. The first dataset is *CiteSeer* [2] dataset, consisting of 3312 papers belonging to 6 different research categories. We consider these categories as the ground truth classes. We construct two layers: *citation layer*, representing the citation relations between papers extracted from the paper citation records; and the paper *similarity layer*, constructed by extracting a vector of 3703 most frequent and unique words for each paper, and then computing the cosine similarity between each pair of papers. The spy plots of the adjacency matrices of these layers are represented in Fig. 4.3 (top). The second dataset is a subset of *CoRA* dataset [271] consisting of 1662 Machine Learning papers grouped into 3 different research categories. Namely, Genetic Algorithms, Neural Networks and Probabilistic Methods. We use the same approach as for *CiteSeer* dataset to construct the *citation* and *similarity* layers. The spy plots of the adjacency matrices of these layers are represented in Fig. 4.3 (bottom). The third dataset we adopt is the MIT Reality Mining Dataset (not shown), which consists of 87 mobile phone users on the MIT campus. The network consist of three layers constructed based on the three types of information: physical location, bluetooth scans and phone calls [260].

Algorithms for clustering multilayer networks

We adopt two baseline approaches, *SNMTF(e)* and *SNMTF(l)*, that are based on SNMTF algorithm. Since *SNMTF* is a single network method, in order to apply it on multilayer networks, we consider the *early* and *late* fusion of network layers respectively. In addition to that, we also consider three state-of-the art techniques, namely Co-regularized Graph Clustering based on NMF (*CGC-NMF*) [262], tensor factorization approach based on PARAFAC decomposition with sparse latent factors (*GraphFuse*) [261] and a spectral method based on the subspace representation of multilayer graphs (*SC-ML*) [257]. Below we briefly explain each of the method and provide the implementation details:

SNMTF(e): we first merge all the network layers into a single network described by the following adjacency matrix: $\mathbf{A} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A}^{(i)}$. Then we apply *SNMTF* on $\mathbf{A}$. To initialize cluster indicator matrix, we use SVD approach for SNMTF [272].

[2] http://linqs.umiacs.umd.edu/projects//projects/lbc/

To extract clusters from the obtained cluster indicator matrix, $\mathbf{H}$, we applied Ncut algorithm [244] on the similarity matrix $\mathbf{H}\mathbf{H}^T$.

SNMTF(l): we first apply *SNMTF* on each network layer, $\mathbf{A}^{(i)}$, separately, and obtain clustering assignment. Then we merge obtained individual clustering assignments into a consensus clustering assignment. We use the same strategy for initialization and clustering extraction as in the case of *SNMTF(e)*.

CGC-NMF: Co-regularized multi-domain Graph Clustering based on symmetric NMF proposed in [262]. In CGC-NMF each domain represents a single network layer. There is a set of parameters $\gamma_{ij} \geq 0$ that balance between single-domain and cross-domain clustering objective for each pair of layers ij. Given that in all our experiments the relationship between node labels for any pair of layers is *one-to-one*, we set $\gamma_{ij} = 1$ (as in [262]) for all pairs of layers and throughout all our experiments. We use the result of SNMTF applied on each network layer separately to initialize the cluster indicator matrices of network layers in CGC-NMF.

SC-ML: Spectral clustering on Multi-Layer graphs proposed in [257]. In SC-ML there is a single regularization parameter, α , that balances the trade-off between two terms in the SC-ML objective function. In all our experiments we choose the value of α that leads to the best clustering performance. In the case of *CiteSeer, CoRA* and *MIT*, the optimal parameters are $\alpha = 0.1$, 0.1 and 0.2, respectively.

GraphFuse: Tensor factorization approach using PARAFAC decomposition with sparse latent factors proposed in [261]. The single parameter, sparsity penalty factor λ, is chosen to lead to the best clustering performance ($\lambda = 0.10$, 0.01 and $1,000$ for *CiteSeer, CoRA* and *MIT*, respectively.

WSSNMTF and *NG-WSSNMTF*: we use the same strategy for matrix initialization and extraction of clusters as described in *SNMTF(e)*. The maximum number of iterations, $max_iter = 100$, has been proven sufficient for reaching the convergence in all our experiments. The sparsity penalty parameter, η, for both methods, is chosen to lead to the best clustering performance (*WSSNMTF*: $\eta = 1.0$, 0.01 and 1.0); *NG-WSSNMTF*: $\eta = 1.0$, 10.0 and $1,000$ for *CiteSeer, CoRA* and *MIT*, respectively.

4.3.5 Results and discussion

The clustering accuracy of the described methods applied on synthetic multilayer network is shown in Table 4.1, with each column representing a different method. The results are shown in terms of the previously introduced clustering evaluation measures (see Sec. 4.2). For each of the three synthetic network, the best result is

Table 4.1: Clustering accuracy measures for three synthetically generated multilayer networks: *SYNT-1* (top), *SYNT-2* (middle) and *SYNT-MIX* (bottom).

		SNMtF (e)	SNMtF (l)	CGC-NMF	SC-ML	GraphFuse	WSSNMTF	NG-WSSNMTF
SYNT-1	Purity	0.878	0.645	0.798	0.993	0.912	**1.000**	**1.000**
	NMI	0.698	0.166	0.434	0.961	0.717	**1.000**	**1.000**
	ARI	0.705	0.150	0.435	0.983	0.709	**1.000**	**1.000**
SYNT-2	Purity	0.937	0.893	0.892	**1.000**	0.950	**1.000**	**1.000**
	NMI	0.810	0.692	0.699	**1.000**	0.872	**1.000**	**1.000**
	ARI	0.798	0.696	0.674	**1.000**	0.866	**1.000**	**1.000**
SYNT-M	Purity	0.963	0.623	0.663	0.993	0.912	0.998	**1.000**
	NMI	0.885	0.171	0.240	0.966	0.728	0.993	**1.000**
	ARI	0.891	0.178	0.247	0.985	0.707	0.996	**1.000**

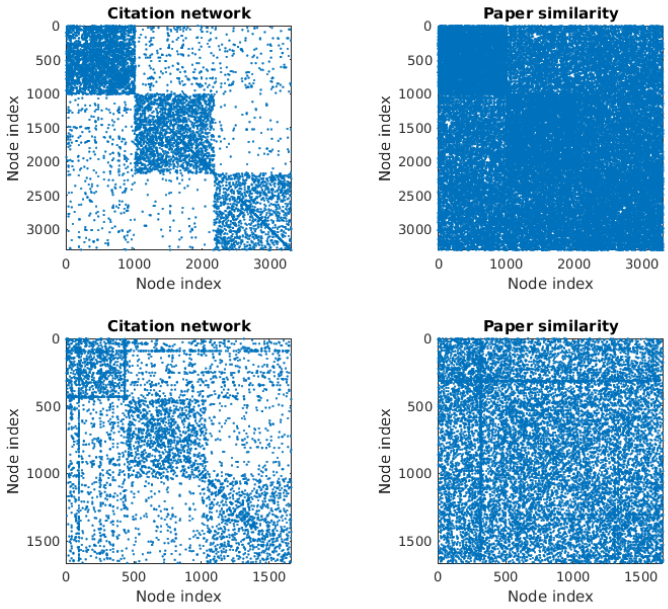


Figure 4.3: (top) *CiteSeer* multilayer network of 3312 papers. Two layers represent citation connections between papers and similarity of papers' 3703 most frequent words, respectively. (bottom) *CoRA* multilayer network of 1662 papers. Two layers represent citation connections between papers and similarity of papers' 1443 most frequent words, respectively. The figure is taken from Gligorijević et al. [227].

highlighted in bold. We observe that both *WSSNMTF* and *NG-WSSNMTF* outperform other methods when applied on *SYNT-1*. Recall that *SYNT-1* is designed to have complementary layers; that is, one community is more present in one layer than in other layers. Given that, we can observe that both of our proposed methods are capable of successfully utilising complementary information contained in all layers and achieving the highest clustering results. Performance of our proposed methods on *SYNT-2*, with layers having different levels of nosy links, is also better than the performance of other factorization methods, and comparable to the performance of *SC-ML* spectral clustering method. On the other hand, baseline methods and other

Table 4.2: Clustering accuracy measures for three real world multilayer networks: *CiteSeer* (top), *CoRA* (middle) and *MIT* (bottom).

		SNMtF (e)	SNMtF (l)	CGC-NMF	SC-ML	GraphFuse	WSSNMTF	NG-WSSNMTF
CiteSeer	Purity	0.558	0.212	0.566	0.630	0.446	0.629	**0.639**
	NMI	0.280	0.014	0.275	0.330	0.159	0.331	**0.337**
	ARI	0.249	0.000	0.248	0.308	0.107	0.316	**0.329**
CoRA	Purity	0.662	0.492	0.687	0.794	0.645	0.796	**0.809**
	NMI	0.345	0.009	0.388	0.496	0.193	0.497	**0.519**
	ARI	0.264	0.001	0.303	0.495	0.187	0.498	**0.526**
MIT	Purity	0.656	0.437	0.663	0.678	0.655	0.575	**0.689**
	NMI	0.440	0.184	0.443	0.460	0.463	0.281	**0.491**
	ARI	0.270	0.037	0.276	0.347	0.316	0.149	**0.395**

factorization methods are characterized with significantly lower clustering performance. The clustering result on the last synthetic network, *SYNT-MIX*, as being the hardest one to cluster, indicate the best performance of our *NG-WSSNMTF*. Comparable results are obtained when using *WSSNMTF* and *SC-ML* methods.

With respect to the real multilayer networks, shown in Table 4.2, both of our methods perform better than other methods on both *CoRA* and *CiteSeer* multilayer networks. Specifically, performance of *NG-WSSNMTF* on *CiteSeer* dataset is slightly better than the performance of *WSSNMTF* and *SC-ML*, but it is significantly better than the performance of other factorization methods. As for *MIT* multilayer networks, performance of both *WSSNMTF* is somehow disappointing in comparison to methods. However, performance of *NG-WSSNMTF* is significantly better than the performance of other methods. On the other hand, note that unlike *CoRA* and *CiteSeer* networks, clustering *MIT* networks is more challenging due to the approximative groundtruth information that is obtained by manual annotation [260].

4.3.6 Conclusion

In this Section, we address the problem of community detection in multilayer networks by proposing two novel methods, *WSSNMTF* and *NG-WSSNMTF*, based on the symmetric non-negative matrix factorization. Both methods work by simultaneously decomposing adjacency matrices representing graph layers while keeping the low-dimensional latent factor matrices the same across the decomposition. In this way both methods ensure to take into account the correlated effect and shared information between network layers. Moreover, both methods are designed to take into account both missing, as well as, noisy links. In terms of clustering accuracy, we demonstrate the superior performance of both of our methods over some base-

line and *state-of-the-art* methods on synthetic data. We show that simple averaging of adjacency matrices representing network layers usually leads to the worst clustering performance. While our first method utilizes ordinary gradient to optimize the objective function, the second method uses natural gradient which forces the common low-dimensional latent matrix factors to be orthonormal which ultimately leads to better clustering results. This has been clearly indicated by significantly better clustering performance on real-world multilayer networks.

4.4 Fusing nonnegative representation of graphs on Grassmann manifold

Here, we improve the previously proposed methods and propose a novel and general model along with its algorithmic framework for clustering multlayer networks. Inspired by recent advances in non-negative matrix factorization (NMF) techniques for graph clustering [251, 252] and by using tools for subspace analysis on Grassmann manifold [257, 269, 270], we propose a general framework for extracting composite communities from multilayer networks. In particular, the framework, termed Network Fusion for Composite Community Extraction (NF-CCE), utilizes four different NMF techniques, each of which is generalized for collective factorization of adjacency matrices representing network layers, and u for computing a consensus low-dimensional latent feature matrix shared across the decomposition. The collective factorization facilitate us to obtain a consensus low-dimensional latent representation, shared across the decomposition, and hence to reveal the communities shared between the network layers. To this end, a general model involving factorization of networks' adjacency matrices and fusion of their low-dimensional subspace representation on Grassmann manifold is proposed in Section 4.4.1. Unlike the methods introduced in the previous Section and a few previously proposed matrix factorization-based methods for multilayer community extraction, e.g., [254, 260, 262, 269], that directly decompose matrices representing network layers into a low-dimensional representation common to all network layers, NF-CCE is conceptually different. Namely, it works in two steps: first, it denoises each network layer by computing its non-negative low-dimensional representation. Then it merges the low-dimensional representations into a consensus low-dimensional representation common to all network layers. This makes our method more robust to noise, and consequently, it yields much stable clustering results.

Four efficient algorithms based on four different NMF techniques are developed for NF-CCE using the concept of *natural gradient* [269, 270] and presented in the form of multiplicative update rules [250] in Section 4.4.1.

The advantages of the NF-CCE over the state-of-the-art in community detection are demonstrated by conducting extensive experiments on a wide range of real-world multilayer networks, including biological, social, economic, citation, phone communication, and brain multilayer networks. In particular, we compared the clustering performance of our four methods with 6 state-of-the-art methods and 5 baseline methods (i.e., single-layer methods modified for multilayer networks), on 9 different real-world multilayer networks including 3 large-scale multiplex biological networks of 3 different species. Experimental results, in Section 4.4.6, indicate that the proposed model yields much stable clustering results than the *state-of-the-art* and baseline methods, by robustly handling incomplete and noisy network layers. The experiments conducted on multilayer biological networks of 3 different species indicate NF-CCE as a superior method for finding composite communities (in biological networks also known as *functional modules*). Moreover, the results also indicate that NF-CCE can extract unique and more functionally consistent modules by considering all network layers together than by considering each network layer separately.

4.4.1 Model setup

Here, we present four novel algorithms that are built upon four non-negative matrix factorization techniques, SNMF [251], PNMF [273], SNMTF [122] and Ssemi-NMTF [274], and extended for fusion and clustering of multilayer networks. Since each of the algorithm is derived in the same way, we present the model using general mathematical notation. We derive a general algorithm, i.e., NF-CCE, that can choose either one of the factorization techniques to extract composite clusters from the collection of network layers. The general algorithm consists of two steps:

- *Step 1* for each network layer, i, we obtain its non-negative, low-dimensional representation, $\mathbf{H}^{(i)}$, under the orthonormality constraint $\mathbf{H}^{(i)T}\mathbf{H}^{(i)} = \mathbf{I}$, by using either one of the non-negative factorization techniques mentioned above.
- *Step 2* we fuse the low-dimensional representations into a common, consensus representation, $\mathbf{H}$, by proposing a collective matrix factorization model. In the model, we collectively decompose all adjacency matrices, $\mathbf{A}^{(i)}$ into a common matrix, $\mathbf{H}$, whilst keeping the non-negative low-dimensional representations of network layers, $\mathbf{H}^{(i)}$ (computed in the previous step), close in distance to the consensus low-dimensional representation, $\mathbf{H}$. The general objective function consisting of these two properties can be written in the following way:

$$\min_{\mathbf{H}\geq 0} \mathcal{J} = \sum_{i=1}^{N} \mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) + \alpha \sum_{i=1}^{N} \mathcal{J}_c^{(i)}(\mathbf{H}; \mathbf{H}^{(i)}) \tag{4.6}$$

where, $\mathcal{J}^{(i)}$ is an objective function for clustering *ith* layer and $\mathcal{J}_c^{(i)}$ is the loss function quantifying the inconsistency between each low-dimensional representation $\mathbf{H}^{(i)}$, computed in *Step 1*, and the consensus representation $\mathbf{H}$.

Below we provide the details of the second step for each individual factorization technique.

Collective SNMF (CSNMF)

We factorize each individual adjacency matrix using Symmetric NMF in the following way:

$$\mathbf{A}^{(i)} \approx \mathbf{H}\mathbf{H}^T$$

under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{H}^T\mathbf{H} = \mathbf{I}$; where, $i = 1, \ldots, N$.

The first part of our general objective function in *Step 2* (Eq. 4.6) has the following form:

$$\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) = \| \mathbf{A}^{(i)} - \mathbf{H}\mathbf{H}^T \|_F^2 \tag{4.7}$$

where $\mathbf{H}$ is the consensus, non-negative low-dimensional matrix, and F denotes Frobenius norm.

Collective PNMF (CPNMF)

We factorize each individual adjacency matrix using Projective NMF in the following way:

$$\mathbf{A}^{(i)} \approx \mathbf{H}\mathbf{H}^T\mathbf{A}^{(i)}$$

under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{H}^T\mathbf{H} = \mathbf{I}$; where, $i = 1, \ldots, N$.

The first part of our general objective function in *Step 2* (Eq. 4.6) has the following form:

$$\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) = \| \mathbf{A}^{(i)} - \mathbf{H}\mathbf{H}^T\mathbf{A}^{(i)} \|_F^2 \tag{4.8}$$

where $\mathbf{H}$ is the consensus, non-negative low-dimensional matrix.

Collective SNMTF (CSNMTF)

We tri-factorize each individual adjacency matrix using Symmetric NMTF in the following way:

$$\mathbf{A}^{(i)} \approx \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T$$

under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{H}^T\mathbf{H} = \mathbf{I}$; where $i = 1, \ldots, N$.

The first part of our general objective function in *Step 2* (Eq. 4.6) has the following form:

$$\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}, \mathbf{S}^{(i)}) = \| \mathbf{A}^{(i)} - \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T \|_F^2 \tag{4.9}$$

where $\mathbf{H}$ is the consensus low-dimensional matrix.

In the derivation of our algorithm, we distinguish between two cases. In the first case we consider $\mathbf{S}$ matrix to be non-negative, i.e., $\mathbf{S}^{(i)} \geq 0$. We call that case CSNMTF. In the second case, we consider $\mathbf{S}^{(i)}$ matrix to have both positive and negative entries. We call this case collective symmetric semi-NMTF, or CSsemi-NMTF (or CSsNMTF).

4.4.2 Merging low-dimensional representation of graph layers on Grassmann Manifolds

For the second term in our general objective function (Eq. 4.6) in *Step 2*, we utilize the orthonormal property of non-negative, low-dimensional matrices, $\mathbf{H}^{(i)}$, and propose a distance measure based on this property. Namely, Dong *et al.* [257] proposed to use the tools from subspace analysis on Grassmann manifold. A Grassmann manifold $\mathbb{G}(k, n)$ is a set of k-dimensional linear subspaces in $\mathbb{R}^n$ [257]. Given that, each orthonormal cluster indicator matrix, $\mathbf{H}_i \in \mathbb{R}^{n \times k}$, spanning the corresponding k-dimensional non-negative subspace, $span(\mathbf{H}_i)$ in $\mathbb{R}^n$, is mapped to a unique point on the Grassmann manifold $\mathbb{G}(k, n)$. The geodesic distance between two subspaces, can be computed by projection distance. For example, the square distance between two subspaces, $\mathbf{H}_i$ and $\mathbf{H}_j$, can be computed as follows:

$$d^2_{proj}(\mathbf{H}_i, \mathbf{H}_j) = \sum_{i=1}^{k} sin^2\theta_i = k - \sum_{i=1}^{k} cos^2\theta_i = k - tr(\mathbf{H}_i\mathbf{H}_i^T\mathbf{H}_j\mathbf{H}_j^T)$$

To find a consensus subspace, $\mathbf{H}$, we factorize all the adjacency matrices, $\mathbf{A}^{(i)}$, and minimize the distance between their subspaces and the consensus subspace on Grassmann manifold. Following this approach, we can write the second part of our general objective function in the following way:

$$\begin{aligned} \mathcal{J}_c^{(i)}(\mathbf{H}, \mathbf{H}^{(i)}) &= k - tr(\mathbf{H}\mathbf{H}^T\mathbf{H}^{(i)}\mathbf{H}^{(i)T}) \\ &= \| \mathbf{H}\mathbf{H}^T - \mathbf{H}^{(i)}\mathbf{H}^{(i)T} \|_F^2 \end{aligned} \tag{4.10}$$

4.4.3 Derivation of the general multiplicative update rules

In *Step 1*, we use well-known non-negative factorization techniques, namely SNMF, PNMF, SNMTF and Ssemi-NMTF, for which the update rules for computing low-dimensional non-negative matrices, $\mathbf{H}^{(i)}$, have been provided in the corresponding papers [122, 251, 273, 274], respectively. They are summarized in Table 4.3 (second column). As for the *Step 2*, we derive the update rules for each of the collective factorization techniques presented in the previous Section 4.4.1. The details of the derivation are in Section D.2 in Appendix D. Here we provide a general update rule for Equation 4.6.

Table 4.3: Multiplicative update rules (MUR) for single-layer and multilayer network analysis.

Method	Single-layer MUR	multilayer MUR
SNMF	$\mathbf{H}_{jk}^{(i)} \leftarrow \mathbf{H}_{jk}^{(i)} \circ \frac{\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\right]_{jk}}$	$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \circ \frac{\left[\left(\sum_{i=1}^{N}\mathbf{A}^{(i)}+\frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\right)\mathbf{H}\right]_{jk}}{\left[\mathbf{H}\mathbf{H}^{T}\left(\sum_{i=1}^{N}\mathbf{A}^{(i)}+\frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\right)\mathbf{H}\right]_{jk}}$
PNMF	$\mathbf{H}_{jk}^{(i)} \leftarrow \mathbf{H}_{jk}^{(i)} \circ \frac{\left[\mathbf{A}^{(i)}\mathbf{A}^{(i)T}\mathbf{H}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{A}^{(i)T}\mathbf{H}^{(i)}\right]_{jk}}$	$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \circ \frac{\left[\left(\sum_{i=1}^{N}\mathbf{A}^{(i)}\mathbf{A}^{(i)T}+\alpha\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\right)\mathbf{H}\right]_{jk}}{\left[\mathbf{H}\mathbf{H}^{T}\left(\sum_{i=1}^{N}\mathbf{A}^{(i)}\mathbf{A}^{(i)T}+\alpha\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\right)\mathbf{H}\right]_{jk}}$
SNMTF	$\mathbf{H}_{jk}^{(i)} \leftarrow \mathbf{H}_{jk}^{(i)} \circ \frac{\left[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]_{jk}}$ $\mathbf{S}_{jk}^{(i)} \leftarrow \mathbf{S}_{jk}^{(i)} \circ \frac{\left[\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\right]_{jk}}$	$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \circ \frac{\left[\sum_{i=1}^{N}\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}+\frac{\alpha}{2}\sum_{i=1}^{N}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\right]_{jk}}{\left[\mathbf{H}\mathbf{H}^{T}\left(\sum_{i=1}^{N}\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}+\frac{\alpha}{2}\sum_{i=1}^{N}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\right)\right]_{jk}}$
Ssemi-NMTF	$\mathbf{H}_{jk}^{(i)} \leftarrow \mathbf{H}_{jk}^{(i)} \circ \frac{\left[\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]^{+}+\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]^{-}\right]_{jk}}{\left[\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]^{-}+\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]^{+}\right]_{jk}}$ $\mathbf{S}^{(i)} \leftarrow \left(\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\right)^{-1}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\left(\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\right)^{-1}$	$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\left[\sum_{i=1}^{N}\left[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\right]^{+}+\mathbf{H}\mathbf{H}^{T}\left[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\right]^{-}+\frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\right]_{jk}}{\left[\sum_{i=1}^{N}\left[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\right]^{-}+\mathbf{H}\mathbf{H}^{T}\left(\left[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\right]^{+}+\frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\right)\right]_{jk}}$

We minimize the general objective function shown in Equation 4.6, under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{H}^T\mathbf{H} = \mathbf{I}$. Namely, we adopt the idea from Ding *et al.*[122] to impose orthonormality constraint on $\mathbf{H}$ matrix, i.e., $\mathbf{H}^T\mathbf{H} = \mathbf{I}$; that has been shown to lead to a more rigorous clustering interpretation [122]. Moreover, assignments of network nodes to composite communities can readily be done by examining the entries in rows of $\mathbf{H}$ matrix. Namely, we can interpret matrix $\mathbf{H}^{n\times k}$ as the *cluster indicator matrix*, where the entries in i-th row (after row normalization) can be interpreted as a posterior probability that a node i belongs to each of the k composite communities. In all our experiments, we apply *hard clustering* procedure, where a node is assign to the cluster that has the largest probability value.

Following the same constrained optimization procedures explained in Sections 4.3.2 and 4.3.3, we derive the update rule, for matrix $\mathbf{H}$, for minimizing the objective function (Eq. 4.6). Specifically, we follow the strategy employed in the derivation of NMF [250] to obtain a multiplicative update rule for $\mathbf{H}$ matrix that can be used for finding a local minimum of the optimization problem (Eq. 4.6).

The derivative of the objective function (Eq. 4.6) with respect to $\mathbf{H}$ is as follows:

$$\nabla_{\mathbf{H}}\mathcal{J} = \sum_{i=1}^{N} \nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) - \alpha \sum_{i=1}^{N} \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H} \tag{4.11}$$

where the first term under summation can be decomposed into two non-negative terms, namely:

$$\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)}) = [\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)})]^{+} - [\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)})]^{-}$$

where, $[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)})]^{+} \geq 0$, $[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H}; \mathbf{A}^{(i)})]^{-} \geq 0$ are non-negative terms. Depending on the type of collective factorization technique, represented in the previous Section, the first term represents the derivative of the corresponding objective function. The second term represents a derivative of Equation 4.10 with respect to $\mathbf{H}$.

To incorporate the orthonormality constraint into the update rule, we introduce the concept of *natural gradient* by following the work of Panagakis *et al.* [269]. Namely, we have shown in Section 4.4.2 that columns of $\mathbf{H}$ matrix span a vector subspace known as Grassmann manifold $\mathbb{G}(k, n)$, i.e., $span(\mathbf{H}) \in \mathbb{G}(k, n)$ [269].

Therefore, following the similar procedure as in Section 4.3.3, we define a natural gradient to optimize our objective function (4.6) under the orthornormality constraint. Following Panagakis *et al.* [269], the natural gradient of $\mathcal{J}$ on Grassmann manifold at $\mathbf{H}$ can be written in terms of the ordinary gradient as follows:

$$\widetilde{\nabla}_{\mathbf{H}}\mathcal{J} = \nabla_{\mathbf{H}}\mathcal{J} - \mathbf{H}\mathbf{H}^T\nabla_{\mathbf{H}}\mathcal{J} \tag{4.12}$$

where, $\nabla_{\mathbf{H}}\mathcal{J}$ is the ordinary gradient given in Equation 4.11.

Following the Karush-Kuhn-Tucker (KKT) complementarity condition [130] and preserving the non-negativity of $\mathbf{H}$, the general update rule for $\mathbf{H}$ matrix using the natural gradient is as follows:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \circ \frac{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^-_{jk}}{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^+_{jk}} \tag{4.13}$$

where, "$\circ$" denotes Hadamard product.

$$[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^- = \mathbf{H}\mathbf{H}^T\sum_{i=1}^{N}[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^+ + \sum_{i=1}^{N}[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^- + \alpha\sum_{i=1}^{N}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}$$

$$[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^+ = \mathbf{H}\mathbf{H}^T\sum_{i=1}^{N}[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^- + \sum_{i=1}^{N}[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^+ + \alpha\mathbf{H}\mathbf{H}^T\sum_{i=1}^{N}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}$$

The concrete update rule for each collective factorization method is summarized in Table 4.3 and united within *NF-CCE* Algorithm 3.

4.4.4 Experiments

We test our methods on synthetic, as well as on real-world data. We designed synthetic multilayer networks with clear ground truth information and different properties in terms of noise and complementary information of network layers, using the *planted partition model* described in Section 4.3.4. The goal is to address the robustness of our methods against noise and their ability to handle complementary information contained in layers. The real-world multilayer networks are taken

Algorithm 3 NF-CCE
Input: Adjacency matrices, $\mathbf{A}^{(i)}$ for each networks layer $i \in \{1, \ldots, N\}$; number of clusters k ; parameter α; factorization technique: FACTORIZATION $\in$ {SNMF, PNMF, SNMTF}
Output: Consensus cluster indicator matrix $\mathbf{H}$

switch (FACTORIZATION)
case 'SNMF'**:**
  for $i \in [1, N]$ **do**
    $\mathbf{H}^{(i)} \leftarrow \text{FACTORIZATION}(\mathbf{A}^{(i)}, k)$
  $\mathbf{A}_{avg} \leftarrow \sum_{i=1}^{N} \mathbf{A}^{(i)} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}$
  $\mathbf{H} \leftarrow \text{FACTORIZATION}(\mathbf{A}_{avg}, k)$
case 'PNMF'**:**
  for $i \in [1, N]$ **do**
    $\mathbf{H}^{(i)} \leftarrow \text{FACTORIZATION}(\mathbf{A}^{(i)}, k)$
  $\mathbf{A}_{avg} \leftarrow \sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{A}^{(i)T} + \alpha\mathbf{H}^{(i)}\mathbf{H}^{(i)T}$
  $\mathbf{H} \leftarrow \text{FACTORIZATION}(\mathbf{A}_{avg}, k)$
case 'SNMTF'**:**
  for $i \in [1, N]$ **do**
    $(\mathbf{H}^{(i)}, \mathbf{S}^{(i)}) \leftarrow \text{FACTORIZATION}(\mathbf{A}^{(i)}, k)$
  $\mathbf{H} \leftarrow \text{CSNMTF}(\{\mathbf{A}^{(i)}\}_{i=1}^{N}, \{\mathbf{H}^{(i)}\}_{i=1}^{N}, \{\mathbf{S}^{(i)}\}_{i=1}^{N}, k)$
end switch

from diverse experimental studies to demonstrate the applicability of our methods in a broad spectrum of disciplines. Namely, we consider social and biological networks, networks of mobile phone communications, brain networks and networks constructed from bibliographic data. The biological networks are treated as a special case because of their lack of ground truth clusters. We provide detailed analysis of such networks based on the functional annotations of their nodes, i.e., genes (see Section 4.2). We present results of comparative analysis of our proposed methods against *state-of-the-art* methods described in Section 4.1.2. Specifically, we compare our methods against PMM, SC-ML, SNF, LMF, GraphFuse and CGC-NMF. Moreover, we adopt the following single-layer methods, SNMF, SNMTF, PNMF and MM (modularity maximization) to be our baseline methods.

Synthetic multilayer networks

Using the *planted partition model*, we generate two sets of synthetic multilayer networks. First type, that we denote *SYNTH-C*, is designed to demonstrate complementary information in layers; whereas the second type, that we denote *SYNTH-N*, is designed to demonstrate different levels of noise between communities contained in layers.

SYNTH-C. We generate two-layer multilayer networks with $n = 200$ nodes and $N = 2$ communities with equal number of nodes each. We generate 11 different mul-

tilayer networks with different amounts of information between two layers. Namely, we vary the within-community probability $p_{11} = \{0.05, 0.075, \ldots, 0.3\}$ of the first community of the first layer across different multilayer networks, while fixing the within-community probability of the second community, $p_{22} = 0.2$. In the second layer, we represent the complementary information by fixing the within-community probability of the first community to $p_{11} = 0.2$ and varying within-cluster probability of the second community $p_{22} = \{0.05, 0.075, \ldots, 0.3\}$ across the multilayer networks. For all multilayer networks, we set the same amount of nosy links, by fixing between-community probability to $p_{12} = 0.05$.

SYNTH-N. Similar to *SYNTH-C* we generate two-layer multilayer networks with two communities (n and N are the same as in *SYNTH-C*). We fix the within-community probability of both communities and both layers to $p_{11} = 0.3$ and $p_{22} = 0.3$ across all multilayer networks. We vary the between-community probability $p_{12} = \{0.02, 0.04, \ldots, 0.2\}$ of the first layer, while keeping the between-community probability of the second layer fixed, $p_{12} = 0.02$, across all multilayer networks.

The spy plots of adjacency matrices representing layers of *SYNTH-C* and *SYNTH-N* are given in Section D.1 of Appendix D.

Real-world multilayer networks

Below we provide a brief description of real-world multilayer networks used in our comparative study:

Bibliographic data, CiteSeer: the data are adopted from [271]. The network consist of $N = 3,312$ papers belonging to 6 different research categories, that we grouped into $k = 3$ pairs categories. We consider these categories as the ground truth classes. We construct two layers: citation layer, representing the citation relations between papers extracted from the paper citation records; and the paper similarity layer, constructed by extracting a vector of $3,703$ most frequent and unique words for each paper, and then computing the cosine similarity between each pair of papers. We construct the k-nearest neighbor graph from the similarity matrix by connecting each paper with its 10 most similar papers.

Bibliographic data, CoRA: the data are adopted from [271]. The network consists of $1,662$ machine learning papers grouped into $k = 3$ different research categories. Namely, Genetic Algorithms, Neural Networks and Probabilistic Methods. We use the same approach as for *CiteSeer* dataset to construct the citation and similarity layers.

Mobile phone data (MPD): the data are adopted from [260]. The network consists of $N = 3$ layers representing different mobile phone communications between $n = 87$ phone users on the MIT campus; namely, the layers represent physical location, bluetooth scans and phone calls. The ground truth clusters are known and manually annotated.

Social network data (SND): the data are adopted from [275]. The dataset represents the multilayer social network of a corporate law partnership, consisting of $N = 3$ layers having three types of edges, namely, co-work, friendship and advice. Each layer has $n = 71$ nodes representing employees in a law firm. Nodes have many attributes. We use the location of employees' offices as well as their status in the firm as the ground truth for clusters and perform two different experiments, namely *SND(o)* and *SND(s)* respectively.

Worm Brain Networks (WBN): the data are retrieved from WormAtlas[3], i.e., from the original study of White *et al.* [276]. The network consist of $n = 279$ nodes representing neurons, connected via $N = 5$ different types of links (i.e., layers), representing 5 different types of synapse. We use neuron types as the ground truth clusters.

Word Trade Networks (WTN): the data represents different types of trade relationships (import/export) among $n = 183$ countries in the world [277]. The network consist of 339 layers representing different products (goods). Since, for some products layers are very sparse, we retain the layers having more than $n - 1$ links, which resulted in $N =$ layers. We use geographic regions (continents) of countries and economic trade categories for defining ground truth clusters[4].

Multiplex biological networks

We obtained multiplex biological networks of 3 different species, i.e., human biological network (HBN), yeast biological network (YBN) and mouse biological network (MBN)[5], from the study of Mostafavi and Morris [146]. The network layers are constructed from the data obtained from different experimental studies and from the publicly available databases. The network layers represent different types of interactions between genes[6], including protein interactions, genetic interaction, gene co-expressions, protein localization, disease associations, etc. The number of nodes and layers in each network is summarized in Table 4.4. For each network and its genes, the corresponding GO annotations has also been provided by Mostafavi and

[3]http://www.wormatlas.org/
[4]data about countries are downloaded from http://unctadstat.unctad.org
[5]The network can be retrieved from: http://morrislab.med.utoronto.ca/~sara/SW/
[6]genes and their coded proteins are considered as the same type of nodes in networks layers

Table 4.4: Real-world multilayer networks used for our comparative study. For each multilayer network we provide the number of nodes, n, number of layers, N, and the number of ground truth clusters, k.

Net name	n	N	ground truth	Ref.
CiteSeer	3,312	2	known ($k = 3$)	[271]
CoRA	1,662	2	known ($k = 3$)	[271]
MPD	87	3	known ($k = 6$)	[260]
SND	71	3	known ($k = 3$)	[275]
WBN	279	10	known ($k = 10$)	[276]
WTN	183	14	known ($k = 5$)	[277]
HBN	13,251	8	unknown ($k = 100$)	[146]
YBN	3,904	44	unknown ($k = 100$)	[146]
MBN	21,603	10	unknown ($k = 100$)	[146]

Morris [146]. For details about network statistics and layer representation we refer a reader to Mostafavi and Morris [146].

In Table 4.4 we summarize the important statistics and information of real-world multilayer networks used in our experiments.

4.4.5 Setup for state-of-the-art methods

Each of the *state-of-the-art* method takes as an input parameter the number of clusters k that needs to be known in advance. Also, some of the methods take as input other types of parameters that needs to be determined. To make the comparison fair, below we briefly explain each of the comparing method and provide the implementation and parameter fitting details that we use in all our experiments (for detailed procedure onm parameter fitting refer to Section D.3 in Appendix D.3):

Baseline, single-layer methods (MM, SNMF, PNMF, SNMTF and Ssemi-NMTF). In order to apply them on multilayer network we first merge all the network layers into a single network described by the following adjacency matrix: $\mathbf{A} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A}^{(i)}$.

PMM[254] has a single parameter, ℓ, which represents the number of structural features to be extracted from each network layer. In all our runs, we compare the clustering performance by varying this parameter, but we also noted that the clustering performance does not change significantly when $l \gg k$.

SNF [258] is parameter-free. However, the method prefers data in the kernel matrix. Thus, we use diffusion kernel matrix representation of binary interaction networks as an input to this method.

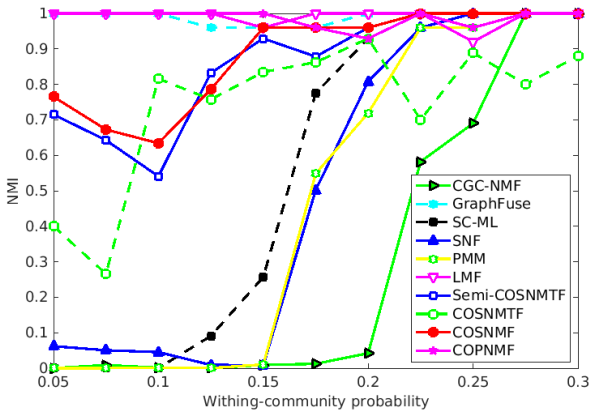


Figure 4.4: The clustering performance of our proposed and other methods on 11 different *SYNTH-C* multilayer networks measured by NMI. On x-axis we present within-community probability, representing the density of connections of communities in the two complementary layers.

SC-ML[257] has a single regularization parameter, α , that balances the trade-off between two terms in the SC-ML objective function. In all our experiments we choose the value of α that leads to the best clustering performance.

LMF[259] has a regularization parameter, α, that balances the influence of regularization term added to objective function to improve numerical stability and avoid over fitting. We vary α in all our runs, and choose the value of α that leads to the best clustering performance.

GraphFuse[261] has a single parameter, sparsity penalty factor λ, that is chosen by exhaustive grid search to lead to the best clustering performance.

CGC-NMF[262] has a set of parameters $\gamma_{ij} \geq 0$ that balance between single-domain and cross-domain clustering objective for each pair of layers ij. Given that in all our experiments the relationship between node labels for any pair of layers is *one-to-one*, we set $\gamma_{ij} = 1$ (as in [262]) for all pairs of layers and throughout all our experiments.

4.4.6 Results and Discussion

Clustering evaluation on artificial multilayer networks

The ability of our proposed methods to extract clusters from complementary layers, represented by *SYNTH-C* networks, is shown in Figure 4.4. The performance of our methods is compared with other methods and it is measured by NMI. By decreasing the within-community probability of complementary clusters in both layer,

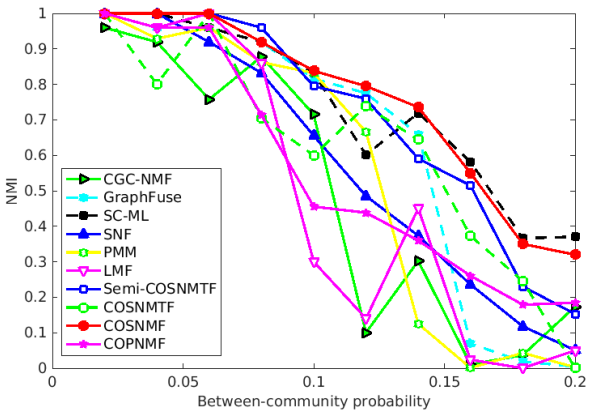


Figure 4.5: The clustering performance of our proposed and other methods on 10 different *SYNTH-N* multilayer networks measured by NMI. On x-axis we present between-community probability, representing the noise level between communities in the first layer.

i.e., by decreasing the density of connections withing communities and thus making communities harder to detect, we see a drastic decrease of performance in many methods, including SC-ML, PMM, SNF and CGC-NMF (Fig. 4.4). Furthermore, below some value of within-community probability, i.e., < 0.1, the performance of these methods is equal or close to zero. Unlike them, our proposed methods, particularly CSNMF, CSsNMTF and CPNMF show significantly better performance. Specifically, CPNMF demonstrates constant performance for all values of withing-community probability. The similar results can also be observed for GraphFuse and LMF. Given that, we can observe that CPNMF method is mostly successful in utilizing complementary information contained in all layers and achieving the highest clustering results.

In terms of noise, incorporated into *SYNTH-N* networks, the ranking between the methods in terms of clustering performance is different. By increasing the between-community probability of the first layer, and thus introducing more noise between communities, the clustering performance of all methods decreases (Fig. 4.5). Our proposed methods, CSNMF, CSNMTF and CSsNMTF, along with SC-ML demonstrate the best performance across different values of within-community probability, which makes them more robust to noise than other methods. On the other hand, other methods methods are characterized with significantly lower clustering performance. Surprisingly, CPNMF method, which is giving the best performance for complementary layers, perform significantly worse on noisy networks than other methods.

Table 4.5: Clustering accuracy of methods (from left to right): MM, SNMF, PNMF, SNMTF, SsNMTF, PMM, SNF, SC-ML, LMF, GraphFuse, CGC-NMF, CSNMF, CPNMF, CSNMTF, CSsNMTF applied on real-world multi-layer networks (from top to bottom): *CiteSeer*, *CoRA*, *MPD*, *SND*, *WBN*, *WTN*, *HBN*, *YBN* and *MBN*.

		MM	SNMF	PNMF	SNMTF	SsNMTF	PMM	SNF	SC-ML	LMF	GF	CGC	CSNMF	CPNMF	CSNMTF	CSsNMTF
CiteSeer	Purity	0.500	0.407	0.405	0.377	0.371	0.302	0.214	0.419	0.235	0.512	0.212	0.501	**0.519**	0.416	0.404
	NMI	0.187	0.222	0.221	0.170	0.164	0.145	0.023	0.191	0.013	0.211	0.013	**0.237**	0.216	0.172	0.195
	ARI	0.152	0.059	0.057	0.042	0.038	0.008	0.001	0.169	0.005	0.201	0.001	**0.207**	0.185	0.093	0.089
CoRA	Purity	0.706	0.669	0.660	0.669	0.669	0.496	0.733	0.787	0.492	0.642	0.678	**0.802**	0.790	0.683	0.684
	NMI	0.340	0.385	0.353	0.382	0.382	0.085	0.449	0.480	0.002	0.201	0.389	**0.514**	0.480	0.346	0.390
	ARI	0.257	0.280	0.247	0.277	0.277	0.030	0.470	0.485	0.001	0.209	0.296	**0.491**	0.470	0.279	0.288
MPD	Purity	0.563	0.678	0.666	0.620	0.678	0.689	0.620	0.701	0.471	0.689	0.678	0.701	0.655	0.655	**0.724**
	NMI	0.313	0.471	0.466	0.384	0.473	0.533	0.395	0.495	0.191	**0.565**	0.457	0.504	0.451	0.458	0.521
	ARI	0.147	0.268	0.259	0.228	0.272	0.383	0.280	0.379	0.029	0.411	0.357	0.394	0.368	0.346	**0.422**
SND(o)	Purity	0.929	**0.943**	**0.943**	0.676	**0.943**	**0.943**	**0.943**	**0.943**	0.788	**0.943**	**0.943**	**0.943**	**0.943**	**0.943**	**0.943**
	NMI	0.618	0.681	0.681	0.133	0.681	0.675	0.689	0.681	0.303	0.675	0.673	0.681	0.685	**0.773**	0.678
	ARI	0.460	0.493	0.493	0.021	0.493	0.477	0.515	0.493	0.239	0.477	0.472	0.493	0.503	**0.811**	0.484
SND (s)	Purity	0.619	0.577	0.633	0.634	0.619	0.591	0.633	0.591	0.633	0.619	0.662	0.633	0.633	**0.747**	0.605
	NMI	0.038	0.025	0.052	0.055	0.041	0.037	0.057	0.030	0.053	0.045	0.0781	0.053	0.053	**0.276**	0.034
	ARI	0.024	0.012	0.058	0.058	0.043	0.022	0.058	0.021	0.059	0.044	0.092	0.059	0.059	**0.234**	0.031
WBN	Purity	0.473	0.512	0.501	0.476	0.523	0.473	0.534	0.272	0.283	0.509	0.516	**0.577**	0.537	0.548	0.530
	NMI	0.333	0.382	0.400	0.327	0.363	0.373	0.425	0.079	0.098	0.426	0.370	**0.463**	0.432	0.404	0.424
	ARI	0.199	0.226	0.213	0.112	0.180	0.130	0.211	0.001	0.009	0.216	0.211	**0.291**	0.233	0.237	0.225
WTN	Purity	0.506	0.475	0.464	0.388	0.284	0.388	0.289	0.497	0.453	0.415	0.278	**0.579**	0.420	0.371	0.420
	NMI	0.231	0.269	0.242	0.176	0.077	0.205	0.073	0.226	0.191	0.176	0.072	**0.322**	0.172	0.154	0.155
	ARI	0.080	0.114	0.114	0.073	0.001	0.039	0.005	0.133	0.094	0.107	0.002	**0.160**	0.094	0.035	0.088
HBN		0.180	0.325	0.350	0.322	0.326	0.351	0.141	0.266	0.045	0.203	0.320	0.341	**0.364**	0.339	0.342
YBN	R_{avg}	0.027	0.374	0.336	0.372	0.358	0.343	0.163	0.257	0.100	0.114	0.311	**0.383**	0.342	**0.383**	0.381
MBN		0.015	0.416	0.387	**0.462**	0.441	0.355	0.211	0.320	0.180	0.298	0.328	0.422	0.401	0.416	0.433

Clustering evaluation on real-world multilayer networks

Table 4.5 presents the Purity, NMI and ARI of our four proposed collective factorization methods, along with five different baseline methods and six different widely used *state-of-the-art* methods on 9 different real-world networks. The first important observation is that all four proposed collective NMF methods (CSNMF, CPNMF, CSNMTF and CSsemi-NMTF) perform better than their corresponding baseline methods (SNMF, PNMF, SNMTF and Ssemi-NMTF) on all real-world multilayer networks. Thus, the strategy of merging layers into a single layer always leads to underperformance. Moreover, single-layer modularity maximization (MM) algorithm is outperformed by baseline, single-layer NMF methods in almost all real-world networks, except for WTN networks where MM significantly outperforms baseline NMF methods, and SND(o) where MM performs better than SNMF, SNMTF and Ssemi-NMTF, but not better than PNMF. In comparison to the multilayer networks *state-of-the-art* methods (PMM, SNF, SC-ML, LMF, GraphFuse and CGC-NMF), at least one of our proposed methods outperforms them all (in terms of either Purity, NMI or ARI or all three measures) in all real-world multilayer network. Moreover, for example, on MPD network, both CSNMF and CSsemi-NMTF perform better than all other methods, with CSemi-NMTF being the best in terms of Purity and NMI; on SND(s) network, CSNMF, CSNMTF and CSsemi-NMTF perform better than all other methods, with CSNMTF performing the best in terms of all three measures; on WBN network, both CSNMF and CSNMTF perform better than all other methods, with CSNMTF being the best in terms of Purity and ARI, and CSNMF being the best in terms of NMI; on WTN network, CSNMF, CSNMTF and CSsemi-NMTF perform better than other methods, with CSNMF being the best in terms of all three measures.

Clustering evaluation on multiplex biological networks

In table 4.5, we also present the average redundancy (R_{avg}) obtained by clustering multiplex biological networks with our four methods, as well as with *state-of-the-art* and baseline methods. The results, again, indicate the superior performance of our methods over the *state-of-the-art* and baseline methods, except in the case of MBN, where SNMTF, applied on merged network layers, yields the highest redundancy.

Furthermore, we compare the functional consistency of clusters obtain by collective integration of all network layers with the functional consistency of clusters obtained from each individual network layer. We report the results both in terms of redundancy and also in terms of average number of statistically significant GO terms computer over the obtained clusters. Redundancy for all three multilayer

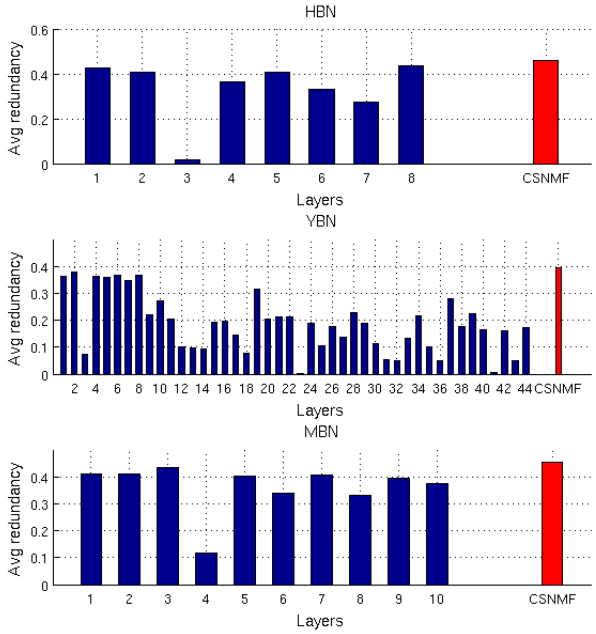


Figure 4.6: Average redundancy of each individual network layer (in blue), computed by *SNMF*, and of their fused representation (in red), computed by *CSNMF*, for networks: (A) HBN, (B) YBN and (C) MBN. Method parameters are: $k = 200$ and $\alpha = 0.01$.

biological networks, obtained by applying SNMF on each individual network layer and CSNMF on all network layers together, is depicted in Fig. 4.6. We observe that each individual network layer has biologically consistent clusters. However, the highest level of biological consistency, measured by the average redundancy, is achieved when all the layers are fused together (red bars in Fig.4.6).

The similar observation stems from Fig. 4.7 that represents the average number of statistically significant GO terms detected by analysing each individual network layer and by fusing all network network layers together. The results indicate that the larger number of significantly enriched GO terms is extracted by integrating all network layers than by considering each network layer separately.

4.4.7 Conclusion

In this Section we address the problem of composite community detection in multi-layer networks by proposing NF-CCE, a general model consisting of four novel methods, *CSNMF*, *CPNMF*, *CSNMTF* and *CSsemi-NMTF*, based on four non-negative matrix factorization techniques. Each of the proposed method works in a similar way: in the first step, it decomposes adjacency matrices representing network layers into low-dimensional, non-negative feature matrices; then, in the second step, it

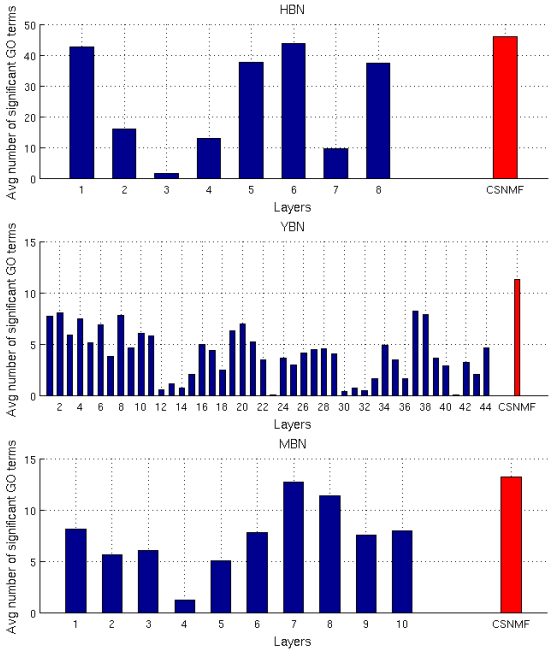


Figure 4.7: Average number of GO terms significantly enriched (i.e., p-val ≤ 0.05) in clusters of each individual network layer (in blue), computed by *SNMF*, and of their fused representation (in red), computed by *CSNMF*, for networks: (A) HBN, (B) YBN and (C) MBN. Method parameters are: $k = 200$ and $\alpha = 0.01$.

fuses the feature matrices of layers into a consensus non-negative, low-dimensional feature matrix common to all network layers, from which the composite clusters are extracted. The second step is done by collective matrix factorization that maximizes the shared information between network layers by optimizing the distance between each of the non-negative feature matrices representing layers and the consensus feature matrix.

The ability of our methods to efficiently integrate complementary as well as noisy network layers has been demonstrated on artificially generated multilayer networks. In terms of clustering accuracy, we demonstrate the superior performance of our proposed methods over the baseline and *state-of-the-art* methods on nine real-world networks. We show that simple averaging of adjacency matrices representing network layers, the strategy usually utilized in the literature, leads to the worst clustering performance. Moreover, our experiments indicate that widely-used modularity maximization methods are significantly outperformed by NMF-based methods.

NF-CCE can be applied on multilayer networks from different domains, ranging from social, phone communication and bibliographic networks to biological, economic and brain networks, demonstrating the diverse applicability of our method.

Chapter 5

Conclusion

In this chapter, we provide a brief summary of our results and contributions in both methodological and biological aspects. We conclude the Thesis by presenting some future directions on how the presented methods can be improved, and by indicating the other biological problems to which they can be applied.

5.1 Summary of Thesis Achievements

Experimental technologies have enabled us to measure and analyse an ever-increasing volume and diversity of biological and biomedical data. A variety of data repositories (reviewed in Section 2.2) containing partially independent information about human genes, proteins, diseases, drugs, patients, etc. are continuously increasing. With an increasing number and heterogeneity of these data available, there is an increasing need for developing efficient and biologically meaningful computational methods for their analysis and integration. Data integration methods (reviewed in Section 2.3) have provided a way of comprehensively and simultaneously analysing data towards more complete understanding of biological and biomedical systems.

In the last two decades, complex networks along with their theoretical foundations have proven to be adequate and very powerful mathematical tool for representing, modeling, and analyzing biological data. However, the multiplicity of biological data sources and their heterogeneous nature have led to the construction of multi-relational networks with heterogeneity reflected in different types of links connecting the same or different types of nodes. In this Theses, in Section 2.1, we have provided one possible classification of such networks. Namely, according to how many types of nodes multi-relational networks have, they can be divided into: *heterogeneous multi-relational networks*, having different types of links connecting

nodes of different types, and *homogeneous multi-relational networks*, having different types of links connecting nodes of the same type. Standard tools for network analysis have proven to be limited in extracting knowledge from such networks, and thus, in this Thesis, motivated by the recent advances in NMF techniques, we have presented novel methods for analysis and integration of multi-relational network data.

In Chapter 3, we propose NMTF method with graph regularization (also denoted as GNMTF) as a general framework for analysis and integration of biological data represented by *heterogeneous multi-relational* network representation. The framework is well suited for simultaneous clustering (co-clustering) of different types of interconnected biological entities, due to its connection to existing clustering techniques; as well as for heterogeneous link prediction in heterogeneous network setup, due to its matrix completion property. The applicability of the proposed framework has been demonstrated on a variety of different biological problems. Particularly, in Section 3.1, we apply the framework on gene annotation prediction and GO reconstruction problems. Using the framework, we integrate four different molecular interaction networks of yeast along with existing gene annotations and GO hierarchy to predict novel gene-GO term annotations and associations between GO terms. Furthermore, we extended the objective function used to model these data by incorporating additional constraints in the form of graphlet-based similarities, to increase the topological information extracted from the molecular interaction networks. We evaluated our predicted associations between GO terms through literature curation, and our novel gene annotations through genetic interaction profiling. Next, in Section 3.2, by further extending the GNMTF technique, we propose a versatile framework for integrating patient-specific somatic mutation data together with molecular networks and drug data, with the goal to address important challenges in precision medicine. Namely, by applying our patient-specific data integration framework on ovarian cancer data, we have identified three groups of patients with significantly different clinical outcomes (measured by survivability analysis). Moreover, our framework can effectively identify novel driver genes responsible for cancer progression, as well as predict drug candidates that could be used for repurposing in the treatment of ovarian cancer. Novel driver gene predictions and drug candidates for repurposing, made by our patient-specific data integration framework, are validated by using external databases that were not included in our study. Possible improvements of the current approach include: 1) integration of additional information on drugs' molecular action, as opposed to chemical structure considered in our study, with a purpose of improving drug repurposing step of our methodology. The key idea is to allow predictions of novel drugs for repurposing that are not only structurally similar with the known drugs already used in the treatment, but also drugs that have similar molecular mechanism of action to drugs that have already

used in the treatment of ovarian cancer. For example, a database such as SIDER [60], which contains data about drug side effects, can be considered as appropriate source of information that can be incorporated into our framework to achieve this goal; 2) including only meaningful and cancer-related patient mutations. Namely, including all possible genes found to be mutated in ovarian cancer introduces a lot of noise. Thus, filtering out genes that have been shown to be mutated in healthy populations, might reduce the noise and consequently lead to more meaningful drug-target predictions. Finally, in Section 3.3, introduce the most general extension of GNMTF technique, and propose Fuse, a novel multiple network alignment method. Fuse can effectively integrate PPI networks of various species, along with sequence similarity computed between proteins of different species and extract clusters of proteins that are functionally and structurally conserved across different species. We demonstrate the superior performance of Fuse over existing aligners in terms of the number of extracted clusters covered by all species and in the terms of the number of them that are functionally consistent; this stems from the fact that Fuse can integrate sequence and network topology more comprehensively than other aligners. Using Fuse, we have also predicted novel clusters of proteins that could be used for transfer of annotations between species (e.g., withing the same identified cluster transfer of annotation from annotated protein of one species to unannotated protein of another species). Possible improvements to our approach include integration of other types of molecular networks that contain complementary information to PPI network. This will require methodological improvements of the existing framework to handle additional layers of information of each species. Thus, instead of aligning PPI networks, the improved framework will allow alignment of more informative, integrated networks of species.

In Chapter 4, we propose two novel types of methods for analysis and integration of data represented by *homogeneous multi-relational* networks (also referred in this Thesis as *multilayer* networks) with the focus on extracting composite communities from the collection of networks. The main difference between these two types is in the strategy used for collective decomposition of network layers. The first type of methods (namely, *WSSNMTF* and *NG-WSSNMTF*), presented in Section 4.3, directly factorize network layers into a common low-dimensional representation from which the composite communities are extracted; whereas, the second type of methods (namely, the *NF-CCE* framework consisting of four different factorization techniques), presented in Section 4.4, first decomposes each layer separately, then it integrates low-dimensional representations of network layers into a common low-dimensional representation from which the composite communities are extracted. The second type of methods have proven more robust to noisy layers, due to denoising step that is absent in the case of *WSSNMTF* and *NG-WSSNMTF*. We demonstrate the superior performance of our proposed methods in comparison to

other state-of-the-art methods, by conducting extensive experiments on both synthetic and a wide range of real-world multilayer networks, including biological, social, economic, citation, phone communication and brain multilayer networks.

5.2 Future Work

Here, we present three ideas for further improvements of our methods presented in Chapter 3 and 4, as well as some preliminary results on these ideas.

5.2.1 Clustering dynamical multilayer networks

There is a recently increasing interest in developing methods for mining time-evolving network data. Thus, extending the *NF-CCE* for multilayer network analysis to handle dynamical data is one of the possible direction of our future work. Potential application could be in extracting stage-specific and evolving clusters of genes from dynamical multilayer network representing different biological networks of gene interactions (e.g., co-expression and co-methylation networks, etc.) at different stages in cancer.

In particular, the only improvement to the objective function of NF-CCE (Equation 4.6) is an additional *temporal smoothing term*. Furthermore, in this Section, we also provide a framework for tracking time-evolving clusters from dynamical multilayer networks.

The schematic representation of dynamical multilayer networks is shown in Figure 5.1. Each time step, corresponding to a particular stage in the dynamics of a system, is represented by layers of networks representing different types of relations between its components. For example, in cancer research, a tumor can be represented as dynamical system, where at each stage, we could have layers representing different biological networks of gene interactions.

Problem formulation

Dynamical multilayer network represents the most general concept of multilayer networks [46]. A temporal multilayer network can be represented as a set of T multilayer networks having N layers at each time step, $\{G_1^{(i)}(V, E_1^{(i)}, w_1^{(i)}), \ldots, G_T^{(i)}(V, E_T^{(i)}, w_T^{(i)})\}$, where $i = 1, \ldots, N$. We restrict our attention to temporal networks in which only edges in network layers vary in time, $E_t^{(i)}$, whereas, the same set of nodes V is present

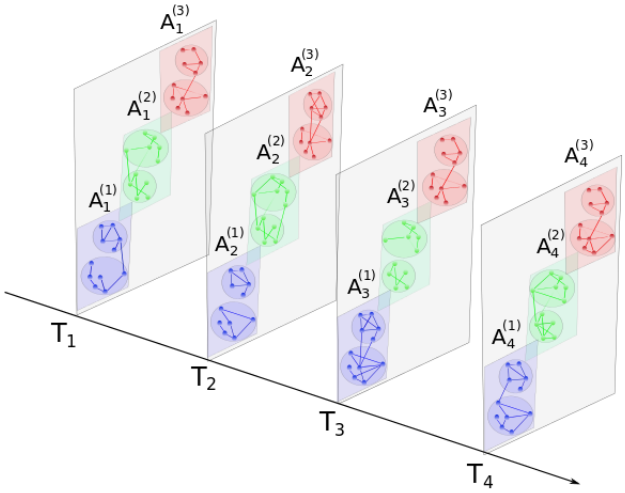


Figure 5.1: An illustration of dynamical multilayer network with $T = 4$ time steps and $N = 3$ layers in each time step.

in all layers and through all time steps. Connectivity patterns of temporal multilayer network can be represented as a set of adjacency matrices, $\vec{\boldsymbol{A}}_t = [\mathbf{A}_t^{(1)}, \ldots, \mathbf{A}_t^{(N)}]$, at each time step $t = 1, \ldots, T$.

Here we define a problem of extracting evolving clusters from temporal multilayer networks, consisting of $t = 1, \ldots, T$ time steps (snapshots) and $i = 1, \ldots, N$ network layers at each time step, using collective Non-negative Matrix Factorization (NMF). Our framework, termed *Evolutionary Collective Orthonormal Symmetric Non-negative Matrix Factorization (ECO-SNMF)*, represents a generalization of our previous method, NF-CCE (with SNMF as the underlying technique for network decomposition), for multilayer networks, extended, here, for time-dependent networks. Namely, building on top of NF-CCE framework, we formulate our a new framework to also satisfy the following two criteria:

- *Network integration:* composite clusters obtained at time t should be consistent with the structure of all network layers, $i = 1, \ldots, N$, at that time step (see Fig. 5.1).
- *Temporal smoothing:* node assignment to clusters at time step t should not differ much from the node assignment to clusters at the previous time step $t - 1$.

Similar to NF-CCE, ECO-SNMF extractins clusters from evolving multilayer networks using the following two steps:

- *Step 1:* find clustering assignment matrices, $\mathbf{G}_t^{(i)}$, for each layer $i = 1, \ldots, N$

and at each time step $t = 1, \ldots, T$:

$$\mathbf{A}_t^{(i)} \approx \mathbf{G}_t^{(i)} \mathbf{G}_t^{(i)T}, \quad \text{where, } \mathbf{G}_t^{(i)T} \mathbf{G}_t^{(i)} = \mathbf{I} \quad \text{and} \quad \mathbf{G}_t^{(i)} \geq 0 \tag{5.1}$$

by solving the following optimization problem $\forall i = 1, \ldots, N$ and $\forall t = 1, \ldots, T$:

$$\min_{\mathbf{G}_t^{(i)} \geq 0} \mathcal{J}(\mathbf{G}_t^{(i)}) = \min_{\mathbf{G}_t^{(i)} \geq 0} \parallel \mathbf{A}_t^{(i)} - \mathbf{G}_t^{(i)} \mathbf{G}_t^{(i)T} \parallel_F^2, \quad \text{s.t.} \quad \mathbf{G}_t^{(t)T} \mathbf{G}_t^{(i)} = \mathbf{I} \tag{5.2}$$

- *Step 2*: find the consensus clustering assignment matrices, $\mathbf{H}_t$, for each time step, $t = 1, \ldots, T$, that satisfy both criteria described above, by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{H}_1 \geq 0, \ldots, \mathbf{H}_T \geq 0} \mathcal{J}(\mathbf{H}_1, \ldots, \mathbf{H}_T) = \min_{\mathbf{H} \geq 0} \Bigg\{ & \sum_{t=1}^{T} \sum_{i=1}^{N} \parallel \mathbf{A}_t^{(i)} - \mathbf{H}_t \mathbf{H}_t^T \parallel_F^2 \\ & + \alpha \sum_{t=1}^{T} \sum_{i=1}^{N} [k - tr(\mathbf{G}_t^{(i)} \mathbf{G}_t^{(i)T} \mathbf{H}_t \mathbf{H}_t^T] \\ & + \beta \sum_{t=1}^{T} [k - tr(\mathbf{H}_{t-1} \mathbf{H}_{t-1}^T \mathbf{H}_t \mathbf{H}_t^T] \Bigg\} \quad \text{s.t. } \forall t, \quad \mathbf{H}_t^T \mathbf{H}_t = \mathbf{I} \end{aligned} \tag{5.3}$$

where, the second term represents the distance between consensus clustering assignment matrix $\mathbf{H}_t$ and each individual clustering assignment matrices $\mathbf{G}_t^{(i)}$ at time step t (fulfilling the first criterion), whereas the third term represents the distance between consensus clustering assignment matrices between time step t and the previous time step, $t-1$ (fulfilling the second criterion). Parameters α and β are trade-off parameters balancing the influence of the second and the third term onto the total cost function, $J(\mathbf{H}_1, \ldots, \mathbf{H}_T)$.

Algorithm for detecting clusters in dynamical multilayer network

To solve the optimization problem given by equation 5.2 in *Step 1*, we use *SNMF* multiplicative update rules (MUR) that are derived in Kuang et al. [251]. Starting with some randomly initialized matrix, $\mathbf{G}_t^{(i)}$, the MUR for updating $\mathbf{G}_t^{(i)}$ matrix are as follows [251]:

$$\mathbf{G}_t^{(i)} \leftarrow \mathbf{G}_t^{(i)} \circ \frac{\mathbf{A}_t^{(i)} \mathbf{G}_t^{(i)}}{\mathbf{G}_t^{(i)} \mathbf{G}_t^{(i)T} \mathbf{A}_t^{(i)} \mathbf{G}_t^{(i)}}, \quad \forall t = 1, \ldots, T, \quad \forall i = 1, \ldots, N \tag{5.4}$$

As for the *Step 2*, we derive the update rules for minimizing the objective function (Equation 5.3). We following the procedure from the constrained optimization theory [130]. Specifically, we follow the strategy employed in the derivation of NMF [250] to obtain a multiplicative update rule for updating each $\mathbf{H}_t$ matrix that can be used for finding a local minimum of the optimization problem.

The derivative of the objective function (Eq. 5.3) with respect to $\mathbf{H}_t$ is as follows:

$$\nabla_{\mathbf{H}_t}\mathcal{J} = -4\sum_{i=1}^{N}\mathbf{A}_t^{(i)}\mathbf{H}_t + 4\sum_{i=1}^{N}\mathbf{H}_t\mathbf{H}_t^T\mathbf{H}_t - 2\alpha\sum_{i=1}^{N}\mathbf{G}_t^{(i)}\mathbf{G}_t^{(i)T}\mathbf{H}_t - 2\beta\mathbf{H}_{t-1}\mathbf{H}_{t-1}^T\mathbf{H}_t \tag{5.5}$$

To incorporate the orthonormality constraint, $\mathbf{H}_t^T\mathbf{H}_t = \mathbf{I}$ into the update rule, we introduce the concept of *natural gradient* by following the work of Panagakis *et al.* [269], who used a *natural gradient* to optimize an objective function under the orthornormality constraint. Following Panagakis *et al.* [269], the natural gradient of $\mathcal{J}$ at $\mathbf{H}_t$ can be written in terms of the ordinary gradient as follows:

$$\widetilde{\nabla}_{\mathbf{H}_t}\mathcal{J} = \nabla_{\mathbf{H}_t}\mathcal{J} - \mathbf{H}_t\mathbf{H}_t^T\nabla_{\mathbf{H}_t}\mathcal{J} \tag{5.6}$$

where, $\nabla_{\mathbf{H}}\mathcal{J}$ is the ordinary gradient given in Equation 5.5.

The natural gradient of our objective function (Eq. 5.3), can be written as follows:

$$\begin{aligned}\widetilde{\nabla}_{\mathbf{H}_t}\mathcal{J} = &-4\sum_{i=1}^{N}\mathbf{A}_t^{(i)}\mathbf{H}_t - 2\alpha\sum_{i=1}^{N}\mathbf{G}_t^{(i)}\mathbf{G}_t^{(i)T}\mathbf{H}_t - 2\beta\mathbf{H}_{t-1}\mathbf{H}_{t-1}^T\mathbf{H}_t \\ &+ 4\sum_{i=1}^{N}\mathbf{H}_t\mathbf{H}_t^T\mathbf{A}_t^{(i)}\mathbf{H}_t + 2\alpha\sum_{i=1}^{N}\mathbf{H}_t\mathbf{H}_t^T\mathbf{G}_t^{(i)}\mathbf{G}_t^{(i)T}\mathbf{H}_t + 2\beta\mathbf{H}_t\mathbf{H}_t^T\mathbf{H}_{t-1}\mathbf{H}_{t-1}^T\mathbf{H}_t\end{aligned} \tag{5.7}$$

Following the Karush-Kuhn-Tucker (KKT) complementarity condition [130] and preserving the non-negativity of $\mathbf{H}$, the general update rule for $\mathbf{H}_t$ matrix using the natural gradient is as follows:

$$\mathbf{H}_t \leftarrow \mathbf{H}_t \circ \frac{[\widetilde{\nabla}_{\mathbf{H}_t}\mathcal{J}]^-}{[\widetilde{\nabla}_{\mathbf{H}_t}\mathcal{J}]^+} \tag{5.8}$$

where, "$\circ$" denotes Hadamard product. The final update rule, can be written in the following form:

$$\mathbf{H}_t \leftarrow \mathbf{H}_t \circ \frac{\mathbf{A}_t^{avg}\mathbf{H}_t}{\mathbf{H}_t\mathbf{H}_t^T\mathbf{A}_t^{avg}\mathbf{H}_t} \quad \forall i = 1, \ldots, T \tag{5.9}$$

where, $\mathbf{A}_t^{avg} = \sum_{i=1}^{N}\left(\mathbf{A}_t^{(i)} + \frac{\alpha}{2}\mathbf{G}_t^{(i)}\mathbf{G}_t^{(i)T}\right) + \frac{\beta}{2}\mathbf{H}_{t-1}\mathbf{H}_{t-1}^T$, with $\mathbf{H}_0 = \mathbf{0}$ is the boundary case. It is interesting to notice that equation 5.9 has the same mathematical form as equation 5.4.

The algorithm, named Evolutionary Collective Orthonormal Symmetric Non-negative Matrix Factorization (ECO-SNMF), that summarizes both *Step 1* and *Step 2* can be represented in Algorithm 4.

Algorithm 4 ECO-SNMF
Input: Adjacency matrices, $\mathbf{A}_t^{(i)}, \quad \forall i = 1, \ldots, N, \quad \forall t = 1, \ldots, T$;
number of clusters k; parameters α and β;
Output: Consensus cluster assignment matrices $\mathbf{H}_t, \quad \forall t = 1, \ldots, T$

for $t \in \{1, \ldots, T\}$ **do**
  for $i \in \{1, \ldots, N\}$ **do**
    $\mathbf{G}_t^{(i)} \leftarrow \text{SNMF}(\mathbf{A}_t^{(i)}, k)$
$\mathbf{A}_1^{avg} = \sum_{i=1}^{N}\left(\mathbf{A}_1^{(i)} + \frac{\alpha}{2}\mathbf{G}_1^{(i)}\mathbf{G}_1^{(i)T}\right)$
$\mathbf{H}_1 \leftarrow SNMF(\mathbf{A}_1^{avg}, k)$
for $t \in \{2, \ldots, T\}$ **do**
  $\mathbf{A}_t^{avg} = \sum_{i=1}^{N}\left(\mathbf{A}_t^{(i)} + \frac{\alpha}{2}\mathbf{G}_t^{(i)}\mathbf{G}_t^{(i)T}\right) + \frac{\beta}{2}\mathbf{H}_{t-1}\mathbf{H}_{t-1}^T$
  $\mathbf{H}_t \leftarrow SNMF(\mathbf{A}_t^{avg}, k)$

Algorithm for tracking clusters' evolution

After obtaining the integrated and temporally smoothed clusters for each network stage, $t = 1, \ldots, T$, we propose to apply the following approach for tracking the dynamics of clusters. The approach is composed of two steps and it is based on *Netgram* [278], an algorithm for pruning bipartite graphs.

- Step 1: Construct weighted bipartite networks between gene clusters, $\mathcal{C}_{t-1}$ and $\mathcal{C}_t$, representing partitions, obtained at two consecutive stages, $t-1$ and t. Each of the partitions, $\mathcal{C}_t$ is composed of k nodes representing clusters of genes obtained at stage t, $\mathcal{C}_t = \{C_1^{(t)}, \ldots, C_k^{(t)}\}$. The link weights between partitions, $\mathcal{C}_{t-1}$ and $\mathcal{C}_t$, is computed via Jaccard index:

$$w_{ij}^{(t)} = \frac{|C_i^{(t-1)} \cap C_i^{(t)}|}{|C_i^{(t-1)} \cup C_i^{(t)}|} \tag{5.10}$$

- Step 2: Pruning bipartite networks, by removing insignificant links as described in *Netgram* [278].

5.2.2 Inference in multi-relational networks

In the last decade, network inference, i.e., reconstruction of network topology of gene interactions from high-throughput biological data, has become one of the foremost problem in systems biology [15, 16, 279–281]. The major focus has been on understanding regulatory interactions between genes by inferring GRN from gene mRNA expression data [16] (see also Section 1.2). Two major computational approaches for inferring GRN networks from biological data are either based on Bayesian networks [16] or Information theory (e.g., correlation-based approaches) [72, 75]. Besides GRN, much effort has also been devoted to inference of other types of biological networks from data, such as signal transduction networks [282, 283] and metabolic networks [284, 285].

An increasing availability of heterogeneous biological data types has imposed new computational challenges and created new opportunities for developing integrative methods in the network inference problem. Based on our previous studies presented in Chapters 3 and 4, here, we propose a novel approach, termed *Network Inference by Matrix Factorization (NIMF)*, for inferring biological networks using NMF and heterogeneous multi-relational network data representation introduced in Section 2.1. Unlike previous methods that mostly focus on inferring interactions between genes, *NIMF* could also be used for inferring biological networks of associations (interactions) between any type of biological entities represented by nodes, e.g., inferring ON, DTI or GDA networks (see Section 2.2 for abbreviations).

Using heterogeneous multi-relational network representation, we distinguish between two types of network inference:

- *heterogeneous network inference*, i.e., inference of a network comprising two types of nodes (bipartite network). See Fig. 5.2 (A).
- *homogeneous network inference*, i.e., inference of a network comprising one type of nodes (''monopartite" network). See Fig. 5.2 (B).

For example, drug-gene-disease associations could be represented by a heterogeneous multi-relation network having three types of nodes schematically represented in Fig. 5.2; namely, drugs represented by red squares, genes by green triangles and diseases by blue circles. For example, in this setup, the first type of network inference refers to an inference of DTI network (i.e., matrix $\mathbf{R}_{13}$), given drug-disease (i.e.,

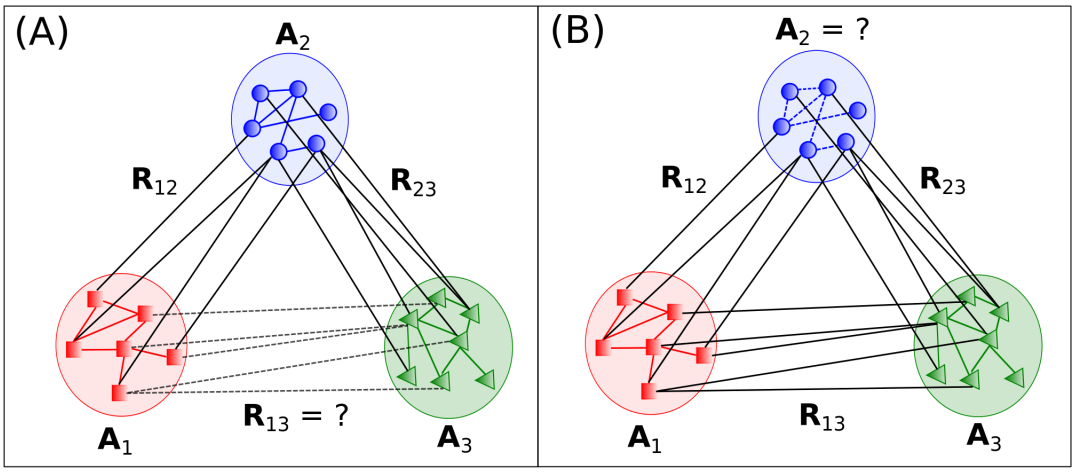


Figure 5.2: Two types of inference in multi-relational networks: (A) inference of bipartite network (associations between red and green nodes) represented by $\mathbf{R}_{13}$ adjacency matrix; (B) inference of ''monopartite" network (associations between blue nodes) represented by $\mathbf{A}_2$ adjacency matrix;

$\mathbf{R}_{12}$ matrix) and gene-disease (i.e., matrix $\mathbf{R}_{23}$) associations, as well as a drug-drug similarity network (i.e., matrix $\mathbf{A}_1$), a gene interaction network (i.e., matrix $\mathbf{A}_3$) and a disease-disease association network ($\mathbf{A}_2$); whereas the second type of inference refers to an inference of disease association network (i.e., matrix $\mathbf{A}_2$), given the same set of adjacency matrices as in the first case including also $\mathbf{R}_{13}$ matrix.

Majority of studies use very simple methods, such as *the projection method* (i.e., collapsing multi-relational network to a single-relational network) to infer networks. For example, in the case of the drug-gene-diseases associations in Fig. 5.2, the interactions between genes and drugs could be inferred in the following way: a gene is a target of a drug (i.e., in Fig. 5.2, DTI link between a red square and a green triangle exists) *iff* the gene and the drug have a disease in common (a blue circle); the more diseases they have in common the link between them is stronger. However, such a simplistic method cannot handle well nosy links that are often present in network layers. Thus, based on NMF and the principle of sharing low-dimensional factors (introduced in Section 2.3), we propose a novel method, i.e., *NIMF*, for inferring networks from multi-relational data. To formulate our model we use the same notation as the one we used in Chapter 3. Also, we formulate the model for a general case of N interconnected networks comprising a multi-relational network, i.e., a network of N different types of nodes (e.g., $N = 3$ in Fig. 5.2), with $\mathbf{R}_{ij}$ representing adjacency matrices of *inter-type* relations between nodes of type i and j, and with $\mathbf{A}_i$ representing adjacency matrices of *intra-type* relations between nodes of type i, where, $i = 1, \ldots, N$.

Problem formulation

Similar to GNMTF, for n_i nodes of type i, we compute their non-negative k-dimensional latent feature vectors represented by matrix $\mathbf{G}_i \in \mathbb{R}_+^{n_i \times k}$ by jointly factorizing all adjacency matrices in the following way:

- adjacency matrices of *intra-type* relations are decomposed by using SNMF: $\mathbf{A}_i \approx \mathbf{G}_i \mathbf{G}_i^T$
- adjacency matrices of *inter-type* relations are decomposed by using NMF: $\mathbf{R}_{ij} \approx \mathbf{G}_i \mathbf{G}_j^T$. The main difference between this decomposition and NMTF, i.e., $\mathbf{R}_{ij} \approx \mathbf{G}_i \mathbf{S}_{ij} \mathbf{G}_j^T$, is in matrix $\mathbf{S}_{ij}$ that in our case is absorbed by factors $\mathbf{G}_i$, and $\mathbf{G}_j$.

After learning low-dimensional latent feature matrices, $\mathbf{G}_i$ from the observed (given) adjacency matrices, the inference of either *inter-type* or *intra-type* relations can be done. To jointly factorize the observed matrices, we formulate the optimization function as follows:

$$\mathcal{J}(\mathbf{G}_1, \ldots, \mathbf{G}_N) = \sum_{i=1}^{N} \sum_{j>i}^{N} \parallel \mathbf{R}_{ij} - \mathbf{G}_i \mathbf{G}_j^T \parallel_F^2 + \frac{\alpha}{2} \sum_{i=1}^{N} \parallel \mathbf{A}_i - \mathbf{G}_i \mathbf{G}_i^T \parallel_F^2 \tag{5.11}$$

where, α is the trade-off parameter which balances the influence of the first and second term; also, the sum goes only over known (i.e., observed) $\mathbf{R}_{ij}$ and $\mathbf{A}_i$ matrices.

To derive the update rules for computing low-dimensional latent feature matrices, we follow the same procedure as in the previous section. We arrive to the following general update rule:

$$\Big(\mathbf{G}_k\Big)_{st} \leftarrow \Big(\mathbf{G}_k\Big)_{st} \circ \frac{\Big[\sum\limits_{i<k}^{N} \mathbf{R}_{ik}^T \mathbf{G}_i + \sum\limits_{i>k}^{N} \mathbf{R}_{ki} \mathbf{G}_i + \alpha \mathbf{A}_k \mathbf{G}_k\Big]_{st}}{\Big[\sum\limits_{i\neq k}^{N} \mathbf{G}_k \mathbf{G}_i^T \mathbf{G}_i + \alpha \mathbf{G}_k \mathbf{G}_k^T \mathbf{G}_k\Big]_{st}} \tag{5.12}$$

where, $k = 1, \ldots, N$. For example, in the case of $N = 3$ interconnected networks (see Fig. 5.2), the corresponding three update rules are as follows:

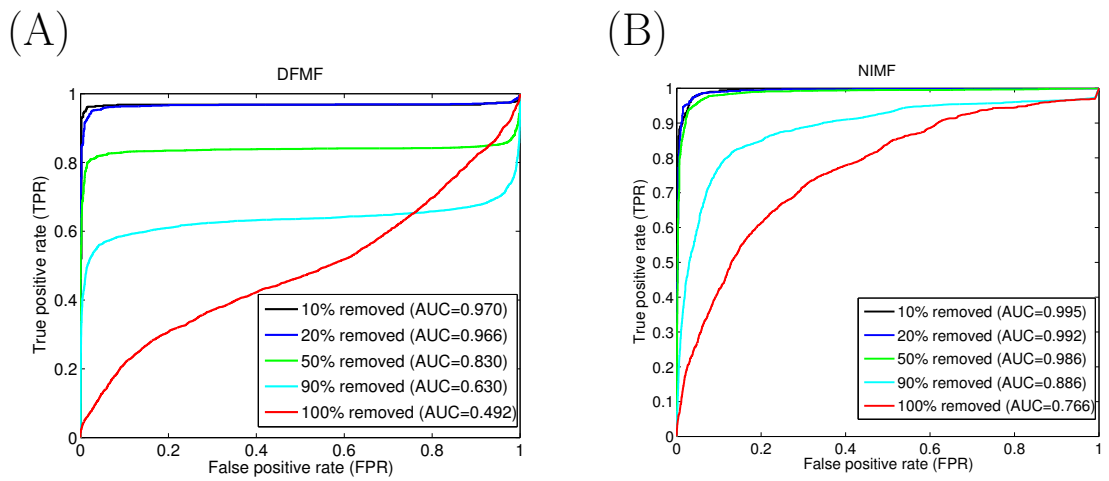


Figure 5.3: ROC curves representing the performance of (A) *DFMF* and (B) *NIMF* in reconstructing $x\%$ of links removed from DTI network. The AUC scores are computed for $x = 10, 20, 50, 90, 100$. The curves represent results averaged over 10 different trials.

$$\begin{aligned}
\Big(\mathbf{G}_1\Big)_{st} &\leftarrow \Big(\mathbf{G}_1\Big)_{st} \circ \frac{\Big[\mathbf{R}_{12}\mathbf{G}_2 + \mathbf{R}_{13}\mathbf{G}_3 + \alpha\mathbf{A}_1\mathbf{G}_1\Big]_{st}}{\Big[\mathbf{G}_1(\mathbf{G}_2^T\mathbf{G}_2 + \mathbf{G}_3^T\mathbf{G}_3 + \alpha\mathbf{G}_1^T\mathbf{G}_1)\Big]_{st}} \\
\Big(\mathbf{G}_2\Big)_{st} &\leftarrow \Big(\mathbf{G}_2\Big)_{st} \circ \frac{\Big[\mathbf{R}_{12}^T\mathbf{G}_1 + \mathbf{R}_{23}\mathbf{G}_3 + \alpha\mathbf{A}_2\mathbf{G}_2\Big]_{st}}{\Big[\mathbf{G}_2(\mathbf{G}_1^T\mathbf{G}_1 + \mathbf{G}_3^T\mathbf{G}_3 + \alpha\mathbf{G}_2^T\mathbf{G}_2)\Big]_{st}} \\
\Big(\mathbf{G}_3\Big)_{st} &\leftarrow \Big(\mathbf{G}_3\Big)_{st} \circ \frac{\Big[\mathbf{R}_{13}^T\mathbf{G}_1 + \mathbf{R}_{23}^T\mathbf{G}_2 + \alpha\mathbf{A}_3\mathbf{G}_3\Big]_{st}}{\Big[\mathbf{G}_3(\mathbf{G}_1^T\mathbf{G}_1 + \mathbf{G}_2^T\mathbf{G}_2 + \alpha\mathbf{G}_3^T\mathbf{G}_3)\Big]_{st}}
\end{aligned} \tag{5.13}$$

Note that the update rules in Equations 5.13 are derived under the assumption that all the adjacency matrices of the multi-relational network shown in Fig. 5.2 are know *a priori*. However, in the heterogeneous network inference problem shown in Fig. 5.2 (A), matrix $\mathbf{R}_{13}$ is unknown. Thus, when using the update rules for computing $\mathbf{G}_i$ matrices, with $i = 1, 2, 3$, the terms involving matrix $\mathbf{R}_{13}$ should be left out from the first and the second update rule. After obtaining $\mathbf{G}_i$ matrices, the DTI adjacency matrix can be inferred from $\mathbf{R}_{13}$ matrix computed as: $\mathbf{R}_{13} = \mathbf{G}_1\mathbf{G}_2^T$. Similarly, in the homogeneous network inference problem shown in Fig. 5.2 (B), matrix $\mathbf{A}_2$ is unknown. Thus, when using the update rules the term in the second update rule involving $\mathbf{A}_2$ matrix should be left out. After obtaining $\mathbf{G}_i$ matrices, disease-disease association network can be inferred from $\mathbf{A}_2$ matrix computed as: $\mathbf{A}_2 = \mathbf{G}_2\mathbf{G}_2^T$.

Our method is mathematically very similar to *DFMF* method [88] introduced in Section 2.3, which is nothing else than collective GNMTF applied on multi-relational networks. However, there are two major differences: 1) the underlying method of *DFMF* is GNMTF that requires computation of $\mathbf{S}_{ij}$ matrices, that in turn require existence of the corresponding $\mathbf{R}_{ij}$ matrices. So, for example, to make a computation of $\mathbf{S}_{13}$ matrix possible, at least a small part of $\mathbf{R}_{13}$ has to be known. Thus, when the whole matrix is missing, the network inference is impossible (see the experiment below); 2) unlike *DFMF* that uses *intra-type* adjacency matrices as constrains in the computation of low-dimensional latent feature matrices, *NIMF* ''learns" *intra-type* adjacency matrices by simultaneously decomposing them together with *inter-type* adjacency matrices, which in turns enables *NIMF* to perform homogeneous network inference.

To test our method and prove its superior performance in comparison to *DFMF*, we conduct an experiment on drug-gene-disease data (as in Fig. 5.2) adopted from Yamanishi et al. [169] [1]. We tested *NIMF* and *DFMF* in the reconstruction of DTI network, by first starting with the whole network as it was known *a priori*. Then, we remove a percentage of known links (i.e., we remove a percentage of observed entries in $\mathbf{R}_{13}$ matrix), and test the performance of both *DFMF* and *NIMF* in their reconstruction. Preliminary results on the performance of the methods for different percentages of removed links are measured by AUC-ROC and are shown in Fig. 5.3. With already 10% of links removed from the DTI network, the *NIMF* (with AUC = 0.995) outperforms *DFMF* (with AUC = 0.970). As the number of removed links increases, the performance of both methods drops with the performance of *DFMF* being lower than the performance of *NIMF*. When the whole DTI network is removed (i.e., when 100% of links are missing from DTI network), *DFMF* is not able to produce any meaningful result (AUC $\lesssim$ 0.5), whereas *NIMF* can still reconstruct the whole DTI network with the accuracy of AUC = 0.766.

From these results, it follows that, using our method, the whole DTI network can be reconstructed from the topology of indirect interactions involving drugs and genes. Furthermore, *NIMF* can be used for predicting new drug-target interactions and thereby postulate for new drug repurposing candidates.

5.2.3 Gene function prediction from multilayer networks

Predicting functions of genes based on their connectivity patterns in a functional interaction network has been a focus of many studies in the last decade [286]. However,

[1] the network are freely available from: http://members.cbio.mines-paristech.fr/~yyamanishi/pharmaco/

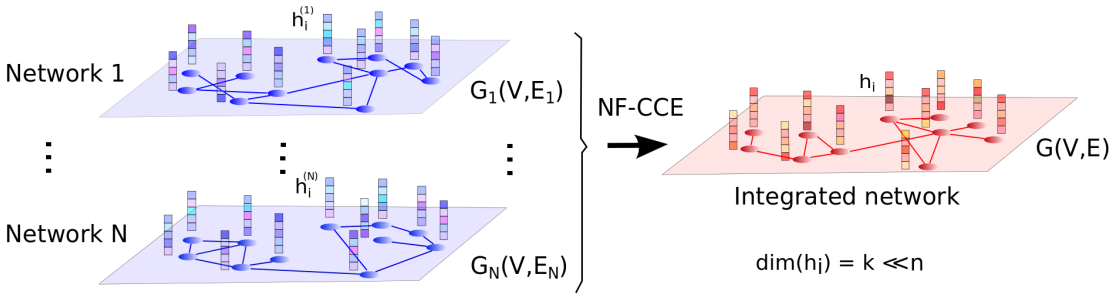


Figure 5.4: Schematic representation of *NF-CCE* framework for combining different networks and computing compact, low-dimensional topological feature representation of genes (denoted by vectors $\mathbf{h}_i$) further used for gene function prediction.

combining different interaction networks constructed from different heterogeneous biological data sources (e.g., genetic interactions, expression data, physical binding, etc.) for the purpose of effective gene function prediction is still an unsolved problem. Developing methods for addressing this problem has become an important challenge in computational biology [147, 287, 288].

Here, we present another view of our network integration framework, *NF-CCE*, introduced in Chapter 4. We propose *NF-CCE* as an underlying approach for gene function classification problem (see Fig. 5.4). Namely, *NF-CCE* could represent a principal framework for combining topological information of genes contained in different molecular interaction networks, and for creating fused compact low-dimensional topological feature representation of genes. Specifically, from each of N different network layers, each having n genes, *NF-CCE* extracts non-negative k-dimensional feature vectors, $\mathbf{h}_i^{(j)}$, where $i = 1, \ldots, n$ and $j = 1, \ldots, N$ (see Fig. 5.4). These vectors represent rows of non-negative cluster indicator matrix $\mathbf{H}^{(j)} = [\mathbf{h}_1^{(j)}, \ldots, \mathbf{h}_n^{(j)}]^T$ (see section 4.4 for notations). Then, *NF-CCE* fuses these vectors into a final non-negative k-dimensional feature vectors, $\mathbf{h}_i$, representing the overall topological information of genes, $i = 1, \ldots, n$, contained in all network layers. These feature vectors could further be used as an input to a multi-label classifier, trained on a known gene-GO annotations. Namely, low-dimensional topological feature vectors could be used for the construction of a kernel matrix, i.e., $K(\mathbf{h}_i, \mathbf{h}_j)$, in the SVM muilti-label classifier [289].

We believe that the gene topological feature representation, obtained by integrating different network with *NF-CCE*, leads to better predictive performance of gene functions that the topological features obtained from any individual network. However, the predictive power of gene topological feature vectors, constructed in this way, is yet to be explored in future work.

Bibliography

[1] L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri *et al.*, "A protein interaction map of Drosophila Melanogaster," *Science*, vol. 302, no. 5651, pp. 1727–1736, 2003.

[2] S. Li, C. Armstrong, N. Bertin *et al.*, "A map of the interactome network of the Metazoan C. elegans," *Science*, vol. 303, no. 5657, pp. 540–543, 2004.

[3] U. Stelzl, U. Worm, M. Lalowski *et al.*, "A human protein-protein interaction network: A resource for annotating the proteome," *Cell*, vol. 122, no. 6, pp. 957 – 968, 2005.

[4] N. Simonis, J.-F. F. Rual, A.-R. R. Carvunis *et al.*, "Empirically controlled mapping of the Caenorhabditis elegans protein-protein interactome network." *Nature Methods*, vol. 6, no. 1, pp. 47–54, 2009.

[5] A. I. M. Consortium, "Evidence for network evolution in an Arabidopsis interactome map," *Science*, vol. 333, no. 6042, pp. 601–607, 2011.

[6] A. Gavin, P. Aloy, P. Grandi *et al.*, "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, no. 7084, pp. 631–636, 2006.

[7] N. Krogan, G. Cagney, H. Yu, G. Zhong *et al.*, "Global landscape of protein complexes in the yeast Saccharomyces Cerevisiae," *Nature*, vol. 440, pp. 637–643, 2006.

[8] R. D. Hawkins, G. C. Hon, and B. Ren, "Next-generation genomics: an integrative approach," *Nature Reviews Genetics*, vol. 11, no. 7, pp. 476–486, 2010.

[9] R. Nielsen, J. S. Paul, A. Albrechtsen, and Y. S. Song, "Genotype and SNP calling from next-generation sequencing data," *Nature Reviews Genetics*, vol. 12, no. 6, pp. 443–451, 2011.

[10] V. Gligorijević and N. Pržulj, "Methods for biological data integration: perspectives and challenges," *Journal of The Royal Society Interface*, vol. 12, no. 112, 2015.

[11] A. R. Joyce and B. Ø. Palsson, "The model organism as a system: integrating *omics* data sets," *Nature Reviews Molecular Cell Biology*, vol. 7, no. 3, pp. 198–210, 2006.

[12] D. Gomez-Cabrero, I. Abugessaisa, D. Maier *et al.*, "Data integration in the era of omics: current and future challenges," *BMC Systems Biology*, vol. 8, no. 2, 2014.

[13] M. Vidal, M. E. Cusick, and A.-L. Barabási, "Interactome Networks and Human Disease," *Cell*, vol. 144, no. 6, pp. 986–998, 2011.

[14] V. Gligorijević, N. Malod-Dognin, and N. Pržulj, "Integrative methods for analyzing big data in precision medicine," *PROTEOMICS*, vol. 16, no. 5, pp. 741–758, 2016.

[15] R. Albert, "Network inference, analysis, and modeling in systems biology," *The Plant Cell*, vol. 19, no. 11, pp. 3327–3338, 2007.

[16] M. Hecker, S. Lambeck, S. Toepfer, E. Van Someren, and R. Guthke, "Gene regulatory network inference: data integration in dynamic models – a review," *Biosystems*, vol. 96, no. 1, pp. 86–103, 2009.

[17] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon *et al.*, "Transcriptional regulatory networks in Saccharomyces cerevisiae," *Science*, vol. 298, no. 5594, pp. 799–804, 2002.

[18] B. Linghu, E. Snitkin, Z. Hu, Y. Xia, and C. DeLisi, "Genome-wide prioritization of disease genes and identification of disease-disease associations from an integrated human functional linkage network," *Genome Biology*, vol. 10, no. 9, p. R91, 2009.

[19] I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte, "A probabilistic functional network of yeast genes," *Science*, vol. 306, no. 5701, pp. 1555–1558, 2004.

[20] B. Rost, J. Liu, R. Nair, K. O. Wrzeszczynski, and Y. Ofran, "Automatic prediction of protein function," *Cellular and Molecular Life Sciences CMLS*, vol. 60, no. 12, pp. 2637–2650, 2003.

[21] M. Ashburner, C. A. Ball, J. A. Blake *et al.*, "Gene ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, pp. 25–29, 2000.

[22] G. R. G. Lanckriet, T. De Bie, N. Cristianini *et al.*, "A statistical framework for genomic data fusion," *Bioinformatics*, vol. 20, no. 16, pp. 2626–2635, 2004.

[23] G. Lanckriet, M. Deng, N. Cristianini, M. Jordan, and W. Noble, "Kernel-based data fusion and its application to protein function prediction in yeast," *Biocomputing 2004, Proceedings of the Pacific Symposium, Hawaii, USA*, pp. 300–311, 2004.

[24] X. Ma, T. Chen, and F. Sun, "Integrative approaches for predicting protein function and prioritizing genes for complex phenotypes using protein interaction networks," *Briefings in Bioinformatics*, 2013.

[25] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein, "A bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae)," *Proceedings of the National Academy of Sciences*, vol. 100, no. 14, pp. 8348–8353, 2003.

[26] V. Gligorijević, V. Janjić, and N. Pržulj, "Integration of molecular network data reconstructs gene ontology," *Bioinformatics*, vol. 30, no. 17, pp. i594–i600, 2014.

[27] N. Nariai, E. D. Kolaczyk, and S. Kasif, "Probabilistic protein function prediction from heterogeneous genome-wide data." *PLoS ONE*, vol. 2, no. 3, pp. e337+, 2007.

[28] H. Wang, H. Huang, C. Ding *et al.*, "Predicting protein–protein interactions from multimodal biological data sources via nonnegative matrix tri-factorization," *Journal of Computational Biology*, vol. 20, no. 4, pp. 344–358, 2013.

[29] T. T. Ashburn and K. B. Thor, "Drug repositioning: identifying and developing new uses for existing drugs," *Nature Reviews Drug Discovery*, vol. 3, no. 8, pp. 673–683, 2004.

[30] F. Napolitano, Y. Zhao, V. M. Moreira *et al.*, "Drug repositioning: A machine-learning approach through data integration." *Journal of Cheminformatics*, vol. 5, p. 30, 2013.

[31] M. A. Hamburg and F. S. Collins, "The path to personalized medicine," *New England Journal of Medicine*, vol. 363, no. 4, pp. 301–304, 2010.

[32] M. D. Ritchie, M. de Andrade, and H. Kuivaniemi, "The foundation of precision medicine: Integration of electronic health records with genomics through basic, clinical, and translational research." *Frontiers in Genetics*, vol. 6, no. 104, 2015.

[33] J. Dudley and K. Karczewski, *Exploring Personal Genomics*. OUP Oxford, 2013.

[34] S. Saria and A. Goldenberg, "Subtyping: What it is and its role in precision medicine," *Intelligent Systems, IEEE*, vol. 30, no. 4, pp. 70–75, 2015.

[35] S. Gutman and L. G. Kessler, "The us food and drug administration perspective on cancer biomarker development," *Nature Reviews Cancer*, vol. 6, no. 7, pp. 565–571, 2006.

[36] Y. Li and S. Jones, "Drug repositioning for personalized medicine," *Genome Medicine*, vol. 4, no. 3, p. 27, 2012.

[37] V. Gligorijević and N. Pržulj, *Computational Methods for Integration of Biological Data*. Cham: Springer International Publishing, 2016, pp. 137–178.

[38] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, September 2000.

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

[40] K. Das, "The laplacian spectrum of a graph," *Computers & Mathematics with Applications*, vol. 48, no. 56, pp. 715 – 724, 2004.

[41] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[42] R. Albert, "Scale-free networks in cell biology," *Journal of Cell Science*, vol. 118, no. 21, pp. 4947–4957, 2005.

[43] A.-L. Barabasi and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, 2004.

[44] N. Pržulj, D. G. Corneil, and I. Jurisica, "Modeling interactome: scale-free or geometric?" *Bioinformatics*, vol. 20, no. 18, pp. 3508–3515, 2004.

[45] D. J. Higham, M. Rašajski, and N. Pržulj, "Fitting a geometric graph to a protein-protein interaction network," *Bioinformatics*, vol. 24, no. 8, pp. 1093–1099, 2008.

[46] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of Complex Networks*, 2014.

[47] D. A. Davis and N. V. Chawla, "Exploring and exploiting disease interactions from multi-relational gene and phenotype networks," *PLoS ONE*, vol. 6, no. 7, p. e22670, 2011.

[48] W. Winterbach, P. V. Mieghem, M. Reinders, H. Wang, and D. d. Ridder, "Topology of molecular interaction networks," *BMC Systems Biology*, vol. 7, no. 1, p. 90, 2013.

[49] A. Chatr-Aryamontri, B.-J. Breitkreutz, S. Heinicke *et al.*, "The BioGRID interaction database," *Nucleic Acids Research*, vol. 41, no. D1, pp. D816–D823, 2013.

[50] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe, "KEGG for integration and interpretation of large-scale molecular data sets," *Nucleic Acids Research*, vol. 40, no. D1, pp. D109–D114, 2012.

[51] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu *et al.*, "DrugBank 3.0: a comprehensive resource for 'omics' research on drugs," *Nucleic Acids Research*, vol. 39, no. suppl 1, pp. D1035–D1041, 2011.

[52] E. E. Bolton, Y. Wang, P. A. Thiessen, and S. H. Bryant, "PubChem: integrated platform of small molecules and biological activities," *Annual Reports in Computational Chemistry*, vol. 4, pp. 217–241, 2008.

[53] A. P. Davis, C. G. Murphy, R. Johnson *et al.*, "The comparative toxicogenomics database: update 2013," *Nucleic Acids Research*, vol. 41, no. D1, pp. D1104–D1114, 2013.

[54] M. J. Li, P. Wang, X. Liu, E. L. Lim, Z. Wang, M. Yeager, M. P. Wong, P. C. Sham, S. J. Chanock, and J. Wang, "GWASdb: a database for human genetic variants identified by genome-wide association studies," *Nucleic Acids Research*, vol. 40, no. D1, pp. D1047–D1054, 2012.

[55] J. C. Denny, L. Bastarache, M. D. Ritchie, R. J. Carroll, R. Zink, J. D. Mosley, J. R. Field, J. M. Pulley, A. H. Ramirez, E. Bowton *et al.*, "Systematic comparison of phenome-wide association study of electronic medical record data and genome-wide association study data," *Nature Biotechnology*, vol. 31, no. 12, pp. 1102–1111, 2013.

[56] L. M. Schriml, C. Arze, S. Nadendla *et al.*, "Disease Ontology: a backbone for disease semantic integration," *Nucleic Acids Research*, vol. 40, no. D1, pp. D940–D946, 2012.

[57] T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, D. Rudnev, C. Evangelista, I. F. Kim, A. Soboleva, M. Tomashevsky, and R. Edgar, "NCBI GEO: mining tens of millions of expression profilesdatabase and tools update," *Nucleic Acids Research*, vol. 35, no. suppl 1, pp. D760–D765, 2007.

[58] H. Parkinson, U. Sarkans, M. Shojatalab *et al.*, "ArrayExpress–a public repository for microarray gene expression data at the EBI," *Nucleic Acids Research*, vol. 33, no. suppl 1, pp. D553–D555, 2005.

[59] J. Hubble, J. Demeter, H. Jin *et al.*, "Implementation of GenePattern within the stanford microarray database," *Nucleic Acids Research*, vol. 37, no. 1, pp. D898–D901, 2009.

[60] M. Kuhn, M. Campillos, I. Letunic, L. J. Jensen, and P. Bork, "A side effect resource to capture phenotypic effects of drugs," *Molecular Systems Biology*, vol. 6, no. 1, 2010.

[61] K. D. Pruitt, T. Tatusova, G. R. Brown, and D. R. Maglott, "NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy," *Nucleic Acids Research*, vol. 40, no. D1, pp. D130–D135, 2012.

[62] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[63] N. Pržulj, "Protein-protein interactions: Making sense of networks via graph-theoretic modeling," *BioEssays*, vol. 33, no. 2, pp. 115–123, 2011.

[64] E. M. Phizicky and S. Fields, "Protein-protein interactions: methods for detection and analysis," *Microbiological Reviews*, vol. 59, no. 1, pp. 94–123, 1995.

[65] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[66] T. Zhou, "Computational reconstruction of metabolic networks from kegg," in *Computational Toxicology*, B. Reisfeld and A. N. Mayeno, Eds. Humana Press, 2013, vol. 930, pp. 235–249.

[67] M. A. Yildirim, K. I. Goh, M. E. Cusick, A. L. Barabási, and M. Vidal, "Drug-target network," *Nature Biotechnology*, vol. 25, no. 10, pp. 1119–1126+, 2007.

[68] M. Costanzo, A. Baryshnikova, J. Bellay *et al.*, "The genetic landscape of a cell," *Science*, vol. 327, no. 5964, pp. 425–431, 2010.

[69] A. Bauer-Mehren, M. Bundschus, M. Rautschka *et al.*, "Gene-disease network analysis reveals functional modules in mendelian, complex and environmental diseases," *PLoS ONE*, vol. 6, no. 6, p. e20284, 2011.

[70] M. Žitnik, V. Janjić, L. Chris *et al.*, "Discovering disease-disease associations by fusing systems-level molecular data," *Scientific Reports*, vol. 3, no. 3202, 2013.

[71] Z. B. Ziv, G. K. Georg, T. I. Tong *et al.*, "Computational discovery of gene modules and regulatory networks," *Nature Biotechnology*, vol. 21, no. 11, pp. 1337–1342, 2003.

[72] P. Langfelder and S. Horvath, "Wgcna: an r package for weighted correlation network analysis," *BMC Bioinformatics*, vol. 9, no. 1, p. 559, 2008.

[73] F. De Smet, J. Mathys, K. Marchal, G. Thijs, B. De Moor, and Y. Moreau, "Adaptive quality-based clustering of gene expression profiles," *Bioinformatics*, vol. 18, no. 5, pp. 735–746, 2002.

[74] F. Luo, Y. Yang, J. Zhong, H. Gao, L. Khan, D. Thompson, and J. Zhou, "Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory," *BMC Bioinformatics*, vol. 8, no. 1, p. 299, 2007.

[75] A. Reverter and E. K. Chan, "Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks," *Bioinformatics*, vol. 24, no. 21, pp. 2491–2497, 2008.

[76] P. Willett, J. M. Barnard, and G. M. Downs, "Chemical similarity searching," *Journal of Chemical Information and Computer Sciences*, vol. 38, no. 6, pp. 983–996, 1998.

[77] N. Nikolova and J. Jaworska, "Approaches to measure chemical similarity–a review," *QSAR & Combinatorial Science*, vol. 22, no. 9-10, pp. 1006–1026, 2003.

[78] P. Zhang, P. Agarwal, and Z. Obradovic, *Computational Drug Repositioning by Ranking and Integrating Multiple Data Sources*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 8190, pp. 579–594.

[79] H. J. Atkinson, J. H. Morris, T. E. Ferrin, and P. C. Babbitt, "Using sequence similarity networks for visualization of relationships across diverse protein superfamilies," *PLoS ONE*, vol. 4, no. 2, p. e4345, 2009.

[80] I. Valavanis, G. Spyrou, and K. Nikita, "A similarity network approach for the analysis and comparison of protein sequence/structure sets," *Journal of Biomedical Informatics*, vol. 43, no. 2, pp. 257–267, 2010.

[81] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, 1990.

[82] L. Holm and J. Park, "DaliLite workbench for protein structure comparison," *Bioinformatics*, vol. 16, no. 6, pp. 566–567, 2000.

[83] H. Ge, A. J. Walhout, and M. Vidal, "Integrating *omic* information: a bridge between genomics and systems biology," *Trends in Genetics*, vol. 19, no. 10, pp. 551 – 560, 2003.

[84] R. Mani, R. P. St.Onge, J. L. Hartman, G. Giaever, and F. P. Roth, "Defining genetic interaction," *Proceedings of the National Academy of Sciences*, vol. 105, no. 9, pp. 3461–3466, 2008.

[85] J. Dutkowski, M. Kramer, M. A. Surma *et al.*, "A gene ontology inferred from molecular networks," *Nature Biotechnology*, vol. 31, no. 1, pp. 38–45, 2013.

[86] O. Gevaert, F. D. Smet, D. Timmerman, Y. Moreau, and B. D. Moor, "Predicting the prognosis of breast cancer by integrating clinical and microarray data with bayesian networks," *Bioinformatics*, vol. 22, no. 14, pp. e184–e190, 2006.

[87] M. H. van Vliet, H. M. Horlings, M. J. van de Vijver, M. J. Reinders, and L. F. Wessels, "Integration of clinical and gene expression data has a synergetic effect on predicting breast cancer outcome," *PLoS ONE*, vol. 7, no. 7, p. e40358, 2012.

[88] M. Žitnik and B. Župan, "Data fusion by matrix factorization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 99, pp. 41–53, 2015.

[89] P. Pavlidis, J. Cai, J. Weston, and W. S. Noble, "Learning gene functional classifications from multiple data types," *Journal of Computational Biology*, vol. 9, pp. 401–411, 2002.

[90] A. Ozen, M. Gonen, E. Alpaydin, and T. Haliloglu, "Machine learning integration for predicting the effect of single amino acid substitutions on protein stability," *BMC Structural Biology*, vol. 9, no. 1, p. 66, 2009.

[91] B. Schölkopf, K. Tsuda, and J.-P. Vert, *Kernel methods in computational biology*. MIT press, 2004.

[92] I. Ben-Gal, *Bayesian Networks*. John Wiley & Sons, Ltd, 2008, ch. 1.

[93] N. Friedman, "Inferring cellular networks using probabilistic graphical models," *Science*, vol. 303, no. 5659, pp. 799–805, 2004.

[94] E. Schadt, S. Friend, and D. Shaywitz, "A network view of disease and compound screening," *Nature Reviews Drug Discovery*, vol. 8, no. 4, pp. 286–295, 2009.

[95] J. Zhu, B. Zhang, E. Smith *et al.*, "Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks," *Nature Genetics*, vol. 40, no. 7, pp. 854–861, 2008.

[96] B. Zhang, C. Gaiteri, L.-G. Bodea, Z. Wang, J. McElwee, A. A. Podtelezhnikov, C. Zhang, T. Xie, L. Tran, R. Dobrin *et al.*, "Integrated systems approach identifies genetic nodes and networks in late-onset alzheimers disease," *Cell*, vol. 153, no. 3, pp. 707–720, 2013.

[97] 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*, vol. 302, no. 5644, pp. 449–453, 2003.

[98] A. Franceschini, D. Szklarczyk, S. Frankild *et al.*, "STRING v9.1: protein-protein interaction networks, with increased coverage and integration," *Nucleic Acids Research*, vol. 41, no. D1, pp. D808–D815, 2013.

[99] Y. Yuan, R. S. Savage, and F. Markowetz, "Patient-specific data fusion defines prognostic cancer subtypes," *PLoS Computational Biology*, vol. 7, no. 10, p. e1002227, 2011.

[100] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[101] C. S. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for SVM protein classification." in *Pacific Symposium on Biocomputing*, 2002, pp. 566–575.

[102] C. S. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble, "Mismatch string kernels for discriminative protein classification," *Bioinformatics*, vol. 20, no. 4, pp. 467–476, 2004.

[103] A. Ben-Hur and D. Brutlag, "Remote homology detection: a motif based approach," *Bioinformatics*, vol. 19, no. suppl 1, pp. i26–i33, 2003.

[104] S. M. Gomez, W. S. Noble, and A. Rzhetsky, "Learning to predict protein–protein interactions from protein sequences," *Bioinformatics*, vol. 19, no. 15, pp. 1875–1881, 2003.

[105] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *In Proceedings of the ICML*, 2002, pp. 315–322.

[106] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, no. 4, pp. 18–28, 1998.

[107] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[108] D. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[109] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[110] K. Borgwardt, "Kernel methods in bioinformatics," in *Handbook of Statistical Bioinformatics*, ser. Springer Handbooks of Computational Statistics, H. H.-S. Lu, B. Scholkopf, and H. Zhao, Eds. Springer Berlin Heidelberg, 2011, pp. 317–334.

[111] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[112] X. Wang, E. P. Xing, and D. J. Schaid, "Kernel methods for large-scale genomic data analysis," *Briefings in Bioinformatics*, p. bbu024, 2014.

[113] S. Yu, L.-C. Tranchevent, B. D. Moor, and Y. Moreau, *Kernel-based Data Fusion for Machine Learning - Methods and Applications in Bioinformatics and Text Mining*, ser. Studies in Computational Intelligence. Springer, 2011, vol. 345.

[114] Y. Wang, S. Chen, N. Deng, and Y. Wang, "Drug repositioning by kernel-based integration of molecular structure, molecular activity, and phenotype data," *PLoS ONE*, vol. 8, no. 11, p. e78518, 11 2013.

[115] A. Daemen, O. Gevaert, and B. De Moor, "Integration of clinical and microarray data with kernel methods," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*. IEEE, 2007, pp. 5411–5415.

[116] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[117] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[118] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science and Business Media, 1992.

[119] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[120] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2005, pp. 606–610.

[121] R. Zass and A. Shashua, "A unifying approach to hard and probabilistic clustering," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 294–301.

[122] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix T-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 126–135.

[123] C. R. Johnson, "Matrix completion problems: a survey," in *Matrix Theory and Applications*, vol. 40. Providence, RI, 1990, pp. 171–198.

[124] S. A. Vavasis, "On the complexity of nonnegative matrix factorization," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1364–1377, 2009.

[125] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[126] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 45–55, 2010.

[127] W. Sun and Y.-X. Yuan, *Optimization theory and methods: nonlinear programming*. Springer Science & Business Media, 2006, vol. 1.

[128] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 556–562.

[129] R. Albright, J. Cox, D. Duling, A. Langville, and C. Meyer, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," *North Carolina State University, Tech. Rep*, vol. 81706, 2006.

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

[131] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. on Knowl. and Data Eng.*, vol. 25, no. 6, pp. 1336–1353, Jun. 2013.

[132] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011.

[133] D. Zhang, S. Chen, and Z.-H. Zhou, "Two-dimensional non-negative matrix factorization for face representation and recognition," in *Proceedings of the Second International Conference on Analysis and Modelling of Faces and Gestures*, ser. AMFG'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 350–363.

[134] S. Zhang, Q. Li, J. Liu *et al.*, "A novel computational framework for simultaneous integration of multiple types of genomic data to identify microrna-gene regulatory modules," *Bioinformatics*, vol. 27, no. 13, pp. i401–i409, 2011.

[135] S. Zhang, C.-C. Liu, W. Li, H. Shen, P. W. Laird, and X. J. Zhou, "Discovery of multi–dimensional modules by integrative analysis of cancer genomic data," *Nucleic Acids Research*, vol. 40, no. 19, p. 93799391, 2012.

[136] C. M. Lee, M. A. V. Mudaliar, D. R. Haggart *et al.*, "Simultaneous nonnegative matrix factorization for multiple large scale gene expression datasets in toxicology," *PLOS ONE*, vol. 7, no. 12, pp. 1–21, 2012.

[137] F. Wang, T. Li, and C. Zhang, "Semi–supervised clustering via matrix factorization," in *SDM*. SIAM, 2008, pp. 1–12.

[138] H. Wang, H. Huang, and C. Ding, "Simultaneous clustering of multi–type relational data via symmetric nonnegative matrix tri-factorization," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, ser. CIKM '11. New York, NY, USA: ACM, 2011, pp. 279–284.

[139] V. Gligorijević, N. Malod-Dognin, and N. Pržulj, *Patient–specific data fusion for cancer stratificaton and personalised treatment*. World Scientific, 2015, pp. 321–332.

[140] V. Gligorijević, N. Malod-Dognin, and N. Pržulj, "Fuse: Multiple network alignment via data fusion," *Bioinformatics*, p. btv731, 2015.

[141] M. Ashburner, C. A. Ball, J. A. Blake *et al.*, "Creating the gene ontology resource: design and implementation," *Genome Research*, vol. 11, no. 8, pp. 1425–1433, 2001.

[142] Y. Chen and D. Xu, "Global protein function annotation through mining genome-scale data in yeast Saccharomyces Cerevisiae," *Nucleic Acids Research*, vol. 32, no. 21, pp. 6414–6424, 2004.

[143] P. Radivojac, W. T. Clark, T. R. Oron *et al.*, "A large-scale evaluation of computational protein function prediction," *Nat Meth*, 2013.

[144] Y. Loewenstein, D. Raimondo, O. Redfern *et al.*, "Protein function annotation by homology-based inference," *Genome Biology*, vol. 10, no. 2, p. 207, 2009.

[145] A. Clare and R. D. King, "Predicting gene function in saccharomyces cerevisiae," *Bioinformatics*, vol. 19, no. 2, pp. ii42–ii49, 2003.

[146] S. Mostafavi and Q. Morris, "Fast integration of heterogeneous data sources for predicting gene function with limited annotation," *Bioinformatics*, vol. 26, no. 14, pp. 1759–1765, 2010.

[147] S. Mostafavi, D. Ray, D. Warde-Farley *et al.*, "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function," *Genome Biology*, vol. 9, no. 1, p. S4, 2008.

[148] M. Žitnik and B. Župan, "Matrix factorization-based data fusion for gene function prediction in baker's yeast and slime mold," in *Biocomputing 2014*, 2014, pp. 400–411.

[149] Y. Liu, Q. Gu, J. Hou *et al.*, "A network-assisted co-clustering algorithm to discover cancer subtypes based on gene expression," *BMC Bioinformatics*, vol. 15, no. 1, pp. 37+, 2014.

[150] T. Hwang, G. Atluri, M. Xie *et al.*, "Co-clustering phenomegenome for phenotype classification and disease gene discovery," *Nucleic Acids Research*, vol. 40, no. 19, p. e146, 2012.

[151] I. Lee, Z. Li, and E. M. Marcotte, "An improved, bias-reduced probabilistic functional gene network of baker's yeast, Saccharomyces Cerevisiae," *PLoS ONE*, vol. 2, no. 10, p. e988, 10 2007.

[152] N. Pržulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, no. 2, pp. e177–e183, 2007.

[153] T. Milenković and N. Pržulj, "Uncovering biological network function via graphlet degree signatures," *Cancer Informatics*, vol. 6, p. 00, 04 2008.

[154] T. Milenković, V. Memišević, A. K. Ganesan *et al.*, "Systems-level cancer gene identification from protein interaction network topology applied to

melanogenesis-related functional genomics data," *Journal of The Royal Society Interface*, vol. 7, no. 44, pp. 423–437, 2010.

[155] F. Shang, L. Jiao, and F. Wang, "Graph dual regularization non-negative matrix factorization for co-clustering," *Pattern Recognition*, vol. 45, no. 6, pp. 2237–2250, 2012.

[156] J.-P. Brunet, P. Tamayo, T. R. Golub *et al.*, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences*, vol. 101, no. 12, pp. 4164–4169, 2004.

[157] A. Sarajlić, J. Vuk, S. Neda *et al.*, "Network topology reveals key cardiovascular disease genes," *PLoS ONE*, vol. 8, no. 8", p. e71537, 08 2013.

[158] D. Zhu, A. O. Hero, H. Cheng *et al.*, "Network constrained clustering for gene microarray data," *Bioinformatics*, vol. 21, no. 21, pp. 4014–4020, 2005.

[159] L. N. Hutchins, S. M. Murphy, P. Singh *et al.*, "Position-dependent motif characterization using non-negative matrix factorization," *Bioinformatics*, vol. 24, no. 23, pp. 2684–2690, 2008.

[160] T. Joshi, Y. Chen, J. M. Becker *et al.*, "Genome-scale gene function prediction using multiple sources of high-throughput data in yeast saccharomyces cerevisiae." *OMICS*, vol. 8, no. 4, pp. 322–33, 2004.

[161] M. Costanzo, B. VanderSluis *et al.*, "A global genetic interaction network maps a wiring diagram of cellular function," *Science*, vol. 353, no. 6306, 2016.

[162] B. Vogelstein, N. Papadopoulos, V. E. Velculescu *et al.*, "Cancer genome landscapes," *Science*, vol. 339, no. 6127, pp. 1546–1558, 2013.

[163] C. Kandoth, M. D. McLellan, F. Vandin, *et al.*, "Mutational landscape and significance across 12 major cancer types," *Nature*, vol. 502, no. 7471, pp. 333–339, 2013.

[164] C. Rubio-Perez, D. Tamborero, M. P. Schroeder *et al.*, "In silico prescription of anticancer drugs to cohorts of 28 tumor types reveals targeting opportunities," *Cancer Cell*, vol. 27, no. 3, pp. 382–396, 2015.

[165] C. G. A. R. Network *et al.*, "Integrated genomic analyses of ovarian carcinoma," *Nature*, vol. 474, no. 7353, pp. 609–615, 2011.

[166] M. Hofree, J. P. Shen, H. Carter, A. Gross, and T. Ideker, "Network-based stratification of tumor mutations," *Nature methods*, vol. 10, no. 11, pp. 1108–1115, 2013.

[167] C. G. A. Network, "Comprehensive molecular characterization of human colon and rectal cancer," *Nature*, vol. 487, no. 7407, pp. 330–337, 2012.

[168] Y. Chen, J. Hao, W. Jiang, T. He, X. Zhang, T. Jiang, and R. Jiang, "Identifying potential cancer driver genes by genomic data integration," *Scientific Reports*, vol. 3, 2013.

[169] Y. Yamanishi, M. Kotera, M. Kanehisa, and S. Goto, "Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework," *Bioinformatics*, vol. 26, no. 12, pp. i246–i254, 2010.

[170] A. Chatr-aryamontri, B.-J. Breitkreutz, R. Oughtred *et al.*, "The biogrid interaction database: 2015 update," *Nucleic Acids Research*, vol. 43, no. D1, pp. D470–D478, 2015.

[171] M. Kanehisa and S. Goto, "KEGG: kyoto encyclopedia of genes and genomes," *Nucleic Acids Research*, vol. 28, no. 1, pp. 27–30, 2000.

[172] D. S. Wishart, C. Knox, A. C. Guo *et al.*, "Drugbank: a knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Research*, vol. 36, no. suppl 1, pp. D901–D906, 2008.

[173] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.

[174] V. Bewick, L. Cheek, and J. Ball, "Statistics review 12: survival analysis," *Critical Care*, vol. 8, pp. 389–394, 2004.

[175] A. P. Davis, C. J. Grondin, K. Lennon-Hopkins, C. Saraceni-Richards, D. Sciaky, B. L. King, T. C. Wiegers, and C. J. Mattingly, "The comparative toxicogenomics database's 10th year anniversary: update 2015," *Nucleic acids research*, vol. 43, no. D1, pp. D914–D920, 2015.

[176] S. Gunther, M. Kuhn, M. Dunkel *et al.*, "Supertarget and matador: resources for exploring drug-target relationships," *Nucleic Acids Research*, vol. 36, no. suppl 1, pp. D919–D922, 2008.

[177] O. Vanunu, O. Magger, E. Ruppin, T. Shlomi, and R. Sharan, "Associating genes and protein complexes with disease via network propagation," *PLoS Computational Biology*, vol. 6, no. 1, p. e1000641, 01 2010.

[178] P. A. Futreal, L. Coin, M. Marshall, T. Down, T. Hubbard, R. Wooster, N. Rahman, and M. R. Stratton, "A census of human cancer genes," *Nature Reviews Cancer*, vol. 4, no. 3, pp. 177–183, 2004.

[179] G. Gundem, C. Perez-Llamas, A. Jene-Sanz *et al.*, "IntOGen: integration and data mining of multidimensional oncogenomic data," *Nature methods*, vol. 7, no. 2, pp. 92–93, 2010.

[180] K. L. Abbott, E. T. Nyre, J. Abrahante, Y.-Y. Ho, R. I. Vogel, and T. K. Starr, "The candidate cancer gene database: a database of cancer driver genes from forward genetic screens in mice," *Nucleic acids research*, vol. 43, no. D1, pp. D844–D848, 2015.

[181] S. Mochizuki and Y. Okada, "Adams in cancer cell proliferation and progression," *Cancer science*, vol. 98, no. 5, pp. 621–628, 2007.

[182] J. Zhao, J. Wang, H. Wang, and M. Lai, "Reg proteins and their roles in inflammation and cancer of the human digestive system," *Adv. Clin. Chem*, vol. 61, pp. 153–173, 2013.

[183] J. Lennartsson and L. Ronnstrand, "The stem cell factor receptor/c-kit as a drug target in cancer," *Current cancer drug targets*, vol. 6, no. 1, pp. 65–75, 2006.

[184] A. Jatoi and C. L. Loprinzi, "Adenosine triphosphate: Does it help cancer patients get bigger and stronger?" *Journal of Clinical Oncology*, vol. 20, no. 2, pp. 362–363, 2002.

[185] E. N. Maldonado and J. J. Lemasters, "Atp/adp ratio, the missed connection between mitochondria and the warburg effect," *Mitochondrion*, vol. 19, pp. 78–84, 2014.

[186] S. I. Deutsch, A. H. Tang, J. A. Burket, and A. D. Benson, "Nmda receptors on the surface of cancer cells: Target for chemotherapy?" *Biomedicine & Pharmacotherapy*, vol. 68, no. 4, pp. 493–496, 2014.

[187] T. Yamakura, K. Sakimura, and K. Shimoji, "N-methyl-d-aspartate receptor channel block by meperidine is dependent on extracellular ph," *Anesthesia & Analgesia*, vol. 90, no. 4, pp. 928–932, 2000.

[188] C. J. Ryan, P. Cimermančič, Z. A. Szpiech, A. Sali, R. D. Hernandez, and N. J. Krogan, "High-resolution network biology: connecting sequence with function," *Nature Reviews Genetics*, 2013.

[189] N. Pržulj, "Protein-protein interactions: Making sense of networks via graph-theoretic modeling," *Bioessays*, vol. 33, no. 2, pp. 115–123, 2011.

[190] K. Mitra, A.-R. Carvunis, S. K. Ramesh, and T. Ideker, "Integrative approaches for finding modular structure in biological networks," *Nature Reviews Genetics*, vol. 14, no. 10, pp. 719–732, 2013.

[191] B. P. Kelley, R. Sharan, R. M. Karp *et al.*, "Conserved pathways within bacteria and yeast as revealed by global protein network alignment," *Proceedings of the National Academy of Sciences*, vol. 100, no. 20, pp. 11 394–11 399, 2003.

[192] O. Kuchaiev, T. Milenković, V. Memišević, W. Hayes, and N. Pržulj, "Topological network alignment uncovers biological function and phylogeny," *Journal of The Royal Society Interface*, vol. 7, no. 50, pp. 1341–1354, 2010.

[193] S. Bandyopadhyay, R. Sharan, and T. Ideker, "Systematic identification of functional orthologs based on protein network comparison," *Genome Research*, vol. 16, no. 3, pp. 428–435, 2006.

[194] P. Uetz, Y.-A. Dong, C. Zeretzke *et al.*, "Herpesviral protein networks and their interaction with the human proteome," *Science*, vol. 311, no. 5758, pp. 239–242, 2006.

[195] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, ser. STOC '71. New York, NY, USA: ACM, 1971, pp. 151–158.

[196] B. P. Kelley, B. Yuan, F. Lewitter *et al.*, "PathBLAST: a tool for alignment of protein interaction networks," *Nucleic Acids Research*, vol. 32, no. suppl 2, pp. W83–W88, 2004.

[197] M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama, "Pairwise alignment of protein interaction networks," *Journal of Computational Biology*, vol. 13, no. 2, pp. 182–199, 2006.

[198] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou, "Graemlin: General and robust alignment of multiple large interaction networks," *Genome Research*, vol. 16, no. 9, pp. 1169–1181, 2006.

[199] R. Singh, J. Xu, and B. Berger, "Pairwise global alignment of protein interaction networks by matching neighborhood topology," in *Research in Computational Molecular Biology*, ser. Lecture Notes in Computer Science, T. Speed and H. Huang, Eds. Springer Berlin Heidelberg, 2007, vol. 4453, pp. 16–31.

[200] C. Clark and J. Kalita, "A comparison of algorithms for the pairwise alignment of biological networks," *Bioinformatics*, 2014.

[201] T. Ito, K. Tashiro, S. Muta *et al.*, "Toward a proteinprotein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins," *Proceedings of the National Academy of Sciences*, vol. 97, no. 3, pp. 1143–1147, 2000.

[202] P. Uetz, L. Giot, G. Cagney *et al.*, "A comprehensive analysis of proteinprotein interactions in saccharomyces cerevisiae," *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.

[203] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader *et al.*, "Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry," *Nature*, vol. 415, no. 6868, pp. 180–183, 2002.

[204] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker, "Conserved patterns of protein interaction in multiple species," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 6, pp. 1974–1979, 2005.

[205] M. Kalaev, M. Smoot, T. Ideker, and R. Sharan, "NetworkBLAST: comparative analysis of protein networks," *Bioinformatics*, vol. 24, no. 4, pp. 594–596, 2008.

[206] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection," *Proceedings of the National Academy of Sciences*, vol. 105, no. 35, pp. 12 763–12 768, 2008.

[207] C.-S. Liao, K. Lu, M. Baym, R. Singh, and B. Berger, "IsorankN: spectral methods for global alignment of multiple protein networks," *Bioinformatics*, vol. 25, no. 12, pp. i253–i258, 2009.

[208] S. M. E. Sahraeian and B.-J. Yoon, "Smetana: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks," *PloS one*, vol. 8, no. 7, p. e67995, 2013.

[209] J. Hu, B. Kehr, and K. Reinert, "Netcoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks," *Bioinformatics*, p. btt715, 2013.

[210] C. Notredame, D. G. Higgins, and J. Heringa, "T-coffee: A novel method for fast and accurate multiple sequence alignment," *Journal of molecular biology*, vol. 302, no. 1, pp. 205–217, 2000.

[211] F. Alkan and C. Erten, "Beams: backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks," *Bioinformatics*, vol. 30, no. 4, pp. 531–539, 2014.

[212] H. Jeong and B.-J. Yoon, "Accurate multiple network alignment through context-sensitive random walk," *BMC systems biology*, vol. 9, no. Suppl 1, p. S7, 2015.

[213] A. Radu and M. Charleston, "Node handprinting: A scalable and accurate algorithm for aligning multiple biological networks," *Journal of Computational Biology*, 2015.

[214] A. Chatr-Aryamontri, B.-J. Breitkreutz, S. Heinicke *et al.*, "The BioGRID interaction database: 2013 update," *Nucleic Acids Research*, vol. 41, no. D1, pp. D816–D823, 2013.

[215] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug 2009.

[216] R. Karp., "Reducibility among combinatorial problems." *Complexity of Computer Computations.*, vol. 6, pp. 85–103, 06 1972.

[217] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, "Entrez gene: gene-centered information at ncbi," *Nucleic Acids Research*, vol. 33, no. suppl 1, pp. D54–D58, 2005.

[218] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, 1990.

[219] M. Žitnik, V. Janjić, C. Larminie, B. Zupan, and N. Pržulj, "Discovering disease–disease associations by fusing systems–level molecular data," *Scientific Reports*, vol. 3, 2013.

[220] G. R. G. Lanckriet, T. De Bie, N. Cristianini *et al.*, "A statistical framework for genomic data fusion," *Bioinformatics*, vol. 20, no. 16, pp. 2626–2635, 2004.

[221] O. Gevaert, F. D. Smet, D. Timmerman, Y. Moreau, and B. D. Moor, "Predicting the prognosis of breast cancer by integrating clinical and microarray data with s," *Bioinformatics*, vol. 22, no. 14, pp. e184–e190, 2006.

[222] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[223] C. H. Papadimitriou, *Computational Complexity*. Addison Wesley Pub. Co., 1994.

[224] G. He, J. Liu, and C. Zhao, "Approximation algorithms for some graph partitioning problems," *Journal of Graph Algorithms and Applications*, vol. 4, no. 2, pp. 1–11, 2000.

[225] J. A. Bondy and U. S. R. Murty, *Graph theory with applications*. Macmillan London, 1976, vol. 6.

[226] L. Lovász and M. Plummer, "Matching theory. number 29 in annals of discrete mathematics," 1986.

[227] V. Gligorijević, Y. Panagakis, and S. Zafeiriou, "Fusion and community detection in multi-layer graphs," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 1327–1332.

[228] ——, "Non-negative matrix factorizations for multiplex network analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. XX, no. XX, pp. XX–XX, 2017.

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

[230] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 45, pp. 175 – 308, 2006.

[231] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, pp. 268–276, 2001.

[232] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.

[233] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[234] J. B. Pereira-Leal, A. J. Enright, and C. A. Ouzounis, "Detection of functional modules from protein interaction networks," *Proteins: Structure, Function, and Bioinformatics*, vol. 54, no. 1, pp. 49–57, 2004.

[235] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 631–640.

[236] J. Chen and B. Yuan, "Detecting functional modules in the yeast protein-protein interaction network," *Bioinformatics*, vol. 22, no. 18, pp. 2283–2290, 2006.

[237] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Phys. Rev. E*, vol. 72, p. 027104, Aug 2005.

[238] 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*, vol. 2008, no. 10, p. P10008, 2008.

[239] M. Mitrović and B. Tadić, "Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities," *Phys. Rev. E*, vol. 80, p. 026123, Aug 2009.

[240] M. A. Porter, J.-P. Onnela, and P. J. Mucha, "Communities in networks," *Notices Amer. Math. Soc.*, vol. 56, no. 9, pp. 1082–1097, 2009.

[241] S. E. Schaeffer, "Survey: Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, Aug. 2007.

[242] J. Kim and J.-G. Lee, "Community detection in multi-layer graphs: A survey," *SIGMOD Rec.*, vol. 44, no. 3, pp. 37–48, dec 2015.

[243] D. B. West *et al.*, *Introduction to graph theory.* Prentice hall Upper Saddle River, 2001, vol. 2.

[244] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[245] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[246] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer, 2001.

[247] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[248] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Networks*, vol. 5, no. 2, pp. 109 – 137, 1983.

[249] M. G. Everett and S. P. Borgatti, "Analyzing clique overlap," *Connections*, vol. 21, no. 1, pp. 49–61, 1998.

[250] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *In NIPS.* MIT Press, 2000, pp. 556–562.

[251] D. Kuang, H. Park, and C. H. Ding, "Symmetric nonnegative matrix factorization for graph clustering." in *SDM*, vol. 12. SIAM, 2012, pp. 106–117.

[252] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, "Community discovery using nonnegative matrix factorization," *Data Min. Knowl. Discov.*, vol. 22, no. 3, pp. 493–521, May 2011.

[253] S. Zhang, H. Zhao, and M. K. Ng, "Functional module analysis for gene co-expression networks with network integration," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 12, no. 5, pp. 1146–1160, Sep. 2015.

[254] L. Tang, X. Wang, and H. Liu, *Uncovering groups via heterogeneous interaction analysis*. IEEE Computer Society, 2009, pp. 503–512.

[255] M. Berlingerio, M. Coscia, and F. Giannotti, "Finding and characterizing communities in multidimensional networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, July 2011, pp. 490–494.

[256] M. A. Rodriguez and J. Shinavier, "Exposing multi–relational networks to single-relational network analysis algorithms," *Journal of Informetrics*, vol. 4, no. 1, pp. 29 – 41, 2010.

[257] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering on multi-layer graphs via subspace analysis on grassmann manifolds," *Trans. Sig. Proc.*, vol. 62, no. 4, pp. 905–918, feb 2014.

[258] B. Wang, A. M. Mezlini, F. Demir *et al.*, "Similarity network fusion for aggregating data types on a genomic scale," *Nature Methods*, vol. 11, no. 3, pp. 333–337, 2014.

[259] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ser. ICDM '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1016–1021.

[260] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with multi-layer graphs: A spectral perspective," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5820–5831, Nov 2012.

[261] E. E. Papalexakis, L. Akoglu, and D. Ienco, "Do more views of a graph help? community detection and clustering in multi-graphs," in *Proceedings of the 16th International Conference on Information Fusion, FUSION 2013, Istanbul, Turkey, July 9-12, 2013*, 2013, pp. 899–905.

[262] W. Cheng, X. Zhang, Z. Guo, Y. Wu, P. F. Sullivan, and W. Wang, "Flexible and robust co-regularized multi-domain graph clustering," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 320–328.

[263] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, no. 1, p. 84, 1970.

[264] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Mach. Learn.*, vol. 55, no. 3, pp. 311–331, jun 2004.

[265] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[266] Y.-K. Shih and S. Parthasarathy, "Identifying functional modules in interaction networks through overlapping markov clustering," *Bioinformatics*, vol. 28, no. 18, pp. i473–i479, 2012.

[267] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists," *Nucleic Acids Research*, vol. 37, no. 1, pp. 1–13, 2009.

[268] Y. Benjamini and Y. Hochberg, "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

[269] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 576–588, March 2010.

[270] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, feb 1998.

[271] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[272] C. Boutsidis and E. Gallopoulos, "Svd based initialization: A head start for nonnegative matrix factorization," *Pattern Recognition*, vol. 41, no. 4, pp. 1350 – 1362, 2008.

[273] Z. Yang and E. Oja, "Linear and nonlinear projective nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 21, no. 5, pp. 734–749, May 2010.

[274] C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, Jan 2010.

[275] T. A. B. Snijders, P. E. Pattison, G. L. Robins, and M. S. Handcock, "New specifications for exponential random graph models," *Sociological Methodology*, vol. 36, no. 1, pp. 99–153, 2006.

[276] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, "The structure of the nervous system of the nematode caenorhabditis elegans," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 314, no. 1165, pp. 1–340, 1986.

[277] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature communications*, vol. 6, p. 6864, 2015.

[278] R. Mall, R. Langone, and J. A. K. Suykens, "Netgram: Visualizing communities in evolving networks," *PLoS ONE*, vol. 10, no. 9, pp. 1–24, 09 2015.

[279] Y. Yamanishi, J.-P. Vert, and M. Kanehisa, "Protein network inference from multiple genomic data: a supervised approach," *Bioinformatics*, vol. 20, no. suppl 1, pp. i363–i370, 2004.

[280] D. Marbach, J. C. Costello, R. Küffner *et al.*, "Wisdom of crowds for robust gene network inference," *Nature Methods*, vol. 9, no. 8, pp. 796–804, jul 2012.

[281] R. Bonneau, D. Reiss, P. Shannon *et al.*, "The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo," *Genome Biology*, vol. 7, no. 5, pp. R36+, 2006.

[282] O. S. Soyer, M. Salath, and S. Bonhoeffer, "Signal transduction networks: Topology, response and biochemical processes," *Journal of Theoretical Biology*, vol. 238, no. 2, pp. 416 – 425, 2006.

[283] M. Steffen, A. Petti, J. Aach, P. D'haeseleer, and G. Church, "Automated modelling of signal transduction networks," *BMC Bioinformatics*, vol. 3, no. 1, p. 34, 2002.

[284] H. Ma and A.-P. Zeng, "Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms," *Bioinformatics*, vol. 19, no. 2, pp. 270–277, 2003.

[285] R. Overbeek, N. Larsen, G. D. Pusch, M. D'Souza, E. Selkov Jr, N. Kyrpides, M. Fonstein, N. Maltsev, and E. Selkov, "WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction," *Nucleic Acids Research*, vol. 28, no. 1, pp. 123–125, 2000.

[286] R. Sharan, I. Ulitsky, and R. Shamir, "Network-based prediction of protein function," *Molecular Systems Biology*, vol. 3, no. 1, 2007.

[287] S. Mostafavi and Q. Morris, "Combining many interaction networks to predict gene function and analyze gene lists," *Proteomics*, vol. 12, no. 10, pp. 1687–1696, 2012.

[288] H. Cho, B. Berger, and J. Peng, "Compact Integration of Multi-Network Topology for Functional Analysis of Genes," *Cell Systems*, vol. 3, no. 6, pp. 540–548.e5, 2016.

[289] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[290] T. E. P. Consortium, "Europe pmc: a full-text literature database for the life sciences and platform for innovation," *Nucleic Acids Research*, vol. 43, no. D1, pp. D1042–D1048, 2015.

[291] J. Masliah-Planchon, L. Darnige, and S. Bellucci, "Molecular determinants of platelet delta storage pool deficiencies: an update," *British journal of haematology*, vol. 160, no. 1, pp. 5–11, 2013.

[292] K. A. Haskins, J. F. Russell, N. Gaddis, H. K. Dressman, and A. Aballay, "Unfolded protein response genes regulated by ced-1 are required for caenorhabditis elegans innate immunity," *Developmental cell*, vol. 15, no. 1, pp. 87–97, 2008.

[293] F. Breinig, T. Sendzik, K. Eisfeld, and M. J. Schmitt, "Dissecting toxin immunity in virus-infected killer yeast uncovers an intrinsic strategy of self-protection," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 10, pp. 3810–3815, 2006.

[294] K. Somogyi, B. Sipos, Z. Pénzes, and I. Andó, "A conserved gene cluster as a putative functional unit in insect innate immunity," *FEBS letters*, vol. 584, no. 21, pp. 4375–4378, 2010.

Appendix A

Important concepts from molecular biology

Here, we present terminology and basic definitions from molecular biology that are used through the thesis.

- *Gene.* A gene is a stretch of DNA encoding functional biomolecules and corresponding to a unit of heredity in living organisms.
- *Allele.* An allele is an alternative form of a gene occupying a given position on a specific chromosome. Individuals inherit two alleles for each gene, one from each parent.
- *Genotype/phenotype.* Genotype/phenotype is an organisms full gene-based hereditary information (excluding only epigenetic information, see below). The phenotype corresponds to the observable properties of an organism, i.e. the expressed hereditary information resulting in a specific morphology, development or behaviour.
- *Genome* is the full set of an organisms hereditary information that is encoded in DNA or in RNA for some viruses. It includes both genetic and non-coding nucleotide sequences, but does not include epigenetic information (see Epigenome below).
- *Epigenome* is the full set of an organisms hereditary information encoded by other structures than genomic DNA and RNA sequences. Epigenetic inheritance mechanisms include DNA methylation and chromatin remodelling (e.g. histone acetylation/de-acetylation), RNA signalling and certain transcription factor activity.

- *Transcriptome* is the full set of all RNA molecules (transcribed from the genes in a genome) in a certain cell type under well-defined conditions. In contrast to the genome, which is mostly static for a given cell line, the transcriptome can change in response to alterations in environmental conditions.

- *Proteome/Metabolome* is the full set of proteins (metabolites) in a given cell type under well–defined conditions. Similar to the transcriptome, the proteome (metabolome) can vary in response to changes in environmental conditions.

- *Protein complex* is a complex containing multiple proteins that (stably) interact with each other.

- *Orthologs* are genes in different species that evolved from a common ancestral gene by speciation (an evolutionary process by which reproductively isolated biological populations evolve to become distinct species).

- *eQTLs* are genomic loci (specific locations on a gene) that regulate expression levels of mRNA. eQTLs that regulate expression of their gene-of-origin are referred as *cis*-eQTL, whereas eQTLs that regulate expression of distant genes are referred as *trans*-eQTL.

- *Copy Number Variations (CNVs)* are regions in the genome having significantly more or less copies than the reference human genome sequence.

- *Methylation* is a process by which methyl groups are added to DNA segments. Methylation changes the activity of a DNA segment without changing the sequence. This is known as an epigenetic modification. When located in a gene promoter, DNA methylation typically acts to repress gene transcription.

- *Model organism* is a species that has been widely studied usually because it is easy to maintain in the laboratory conditions.

Appendix B

Reconstruction of GO by integration of molecular networks

B.1 Supplementary figures

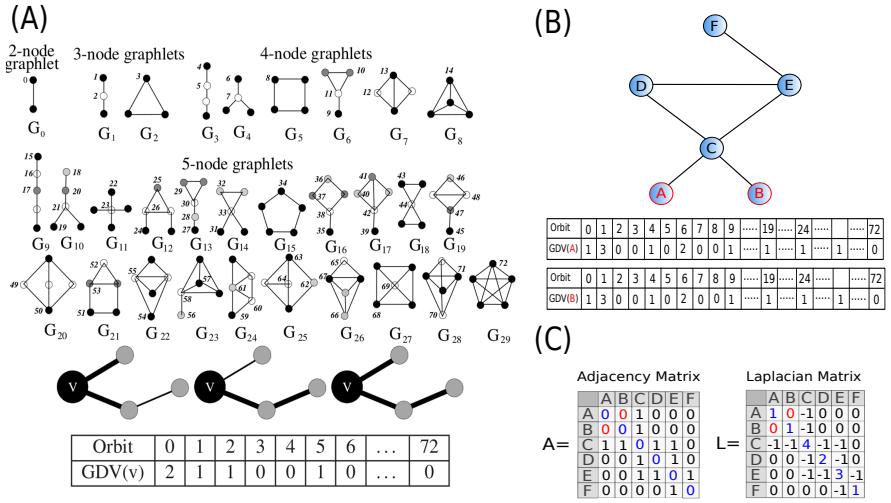


Figure B.1: (A) *Above*: Graphltes containing 2-5 nodes, denoted by $G_0, G_1, \ldots G_{29}$ and their automorphic orbits $i \in \{0, 1, \ldots, 72\}$. Nodes within a graphlet belonging to the same orbit (i.e. invariant under the symmetry group) are represented by the same shade. *Below*: An example of a GDV vector for a node v. The node touches G_0 graphlet (orbit 0) two times; end node of G_1 graphlet (orbit 1) one time and middle node of the same graphlet (orbit 2) one time; and orbit 5 of the graphlet G_3 one time. (B) An example of the graph containing 6 nodes along with the GDV vectors of nodes A and B. Identical GDV vectors of nodes A and B indicate that these two nodes have the same local topology even though they are not directly connected. (C) Adjacency and Laplacian matrices for the graph shown in panel (B). Zer oentries in red indicate that nodes A and B are not connected.

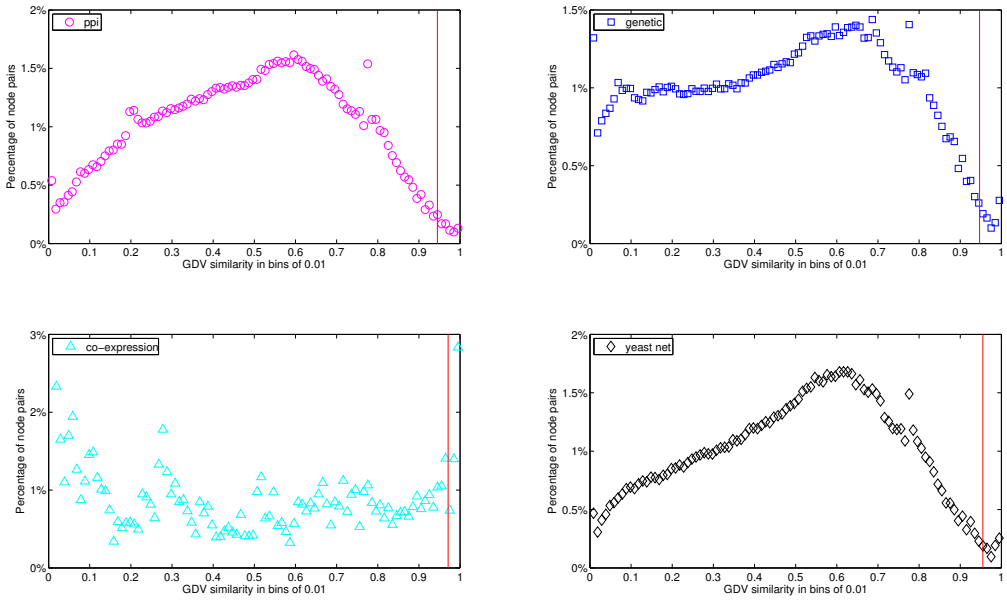


Figure B.2: GDV similarity distributions: Distribution of gene pairs according to their GDV similarity in four different networks of *Saccharomyces cerevisiae*: (A) protein-protein interaction (PPI) network; (B) genetic interaction (GI) network; (C) gene co-expression network; (D) integrated function network (YeastNet). Vertical red line represents a threshold for a given distribution obtained by selecting top 1% of the most similar gene pairs (corresponding to a p-value of 0.01).

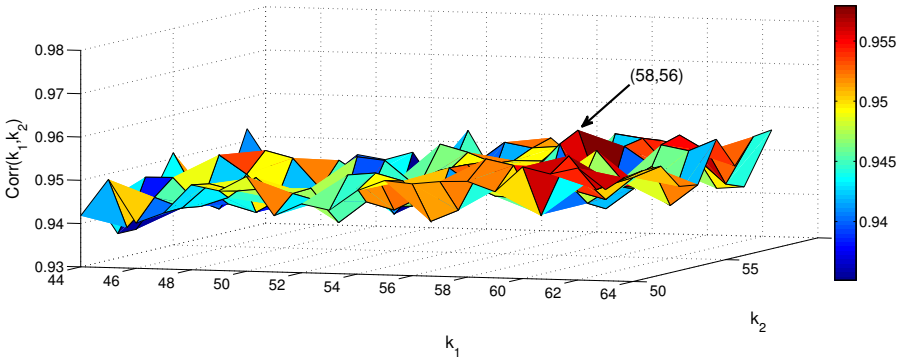


Figure B.3: Estimation of rank parameters by computing cophenetic correlation coefficient: For each pair $(k_1^{(i)}, k_2^{(j)})$ of ranks, depicted in the (x, y) plane we compute averaged cophenetic correlation coefficient for two data sets - *genes* and *GO terms*. Averaged coefficient is represented on the z axis. We identified a set of parameters $(k_1, k_2) = (58, 56)$, as our optimal ranks, for which there is a maximum value of cophenetic correlation coefficient (a peak on this 3D color plot). Around this value, correlation coefficient significantly drops.

Appendix C

Multiple network alignment

C.1 Study of different parameters

Study of different values of parameter γ

To determine the influence of the regularization parameter γ, we perform NMTF runs for $\gamma \in \{0.0, 0.001, 0.01, 0.1, 0.5, 0.7, 0.9\}$. For each run, we evaluate the functional consistency of the protein associations that are predicted by GNMTF (Fig. C.1), as well as the functional consistency of the obtained clusters after the alignment step (Fig. C.2). In both cases, we follow the same methodology as in the main document.

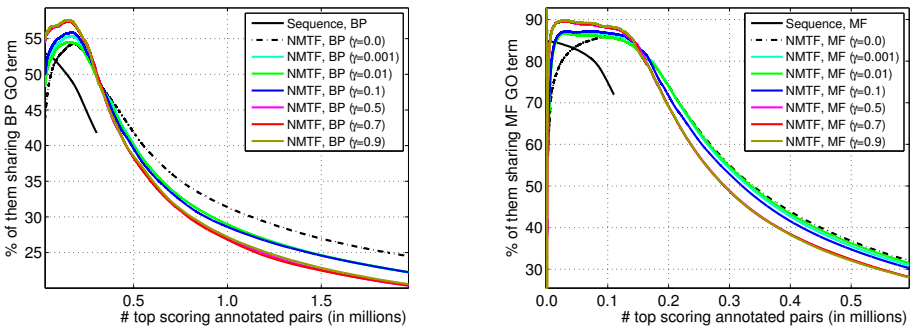


Figure C.1: Functional consistency of GNMTF and sequence-based associations for different values of γ. For all of sequence similarities and NMTF based similarities with different values of γ, we plot the cumulative number of annotated protein pairs (x-axis) against the percentages of them sharing GO terms (y-axis). Biological process (BP) and molecular function (MF) annotations are presented on the left and on the right panels, respectively.

As presented in supplementary Figure C.1, NMTF achieves similar functional

consistencies for $\gamma \in \{0.5, 0.7, 0.9\}$, with $\gamma = 0.7$ produces the highest number of consistent protein pairs with the top association scores. Moreover, the corresponding curves are above the ones corresponding to $\gamma \in \{0.0, 0.001, 0.01, 0.01\}$; this indicates that a larger influence of PPI networks (measured by parameter γ) results in higher functional consistency of the predicted associations between proteins. Finally, after the alignment step, the highest functional consistency of the produced clusters is achieved for $\gamma = 0.7$ (see supplementary Figure C.2). Therefore, in the main document and in the the rest of this supplementary material, we consider the results of Fuse with $\gamma = 0.7$.

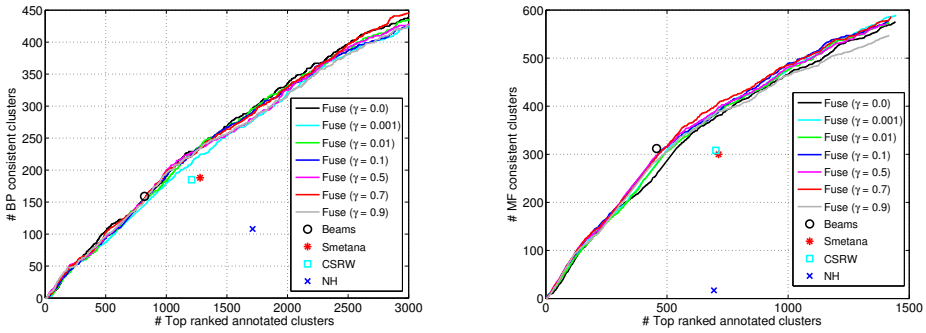


Figure C.2: Functional consistency of Fuse's clusters for different values of γ. For Fuse's protein clusters with different values of γ, we plot the cumulative number of annotated clusters containing proteins from all 5 species (x-axis) against the number of them that are functionally consistent (y-axis). Clusters are ranked according to the scores computed as the sum of association scores between their proteins. We also report the total number of annotated and consistent clusters for the competing aligners (points in the plots). Biological process (BP) and molecular function (MF) annotations are considered separately in the left-hand-side and right-hand-side panels, respectively.

Study of different values of parameter α

To estimate the influence of parameter α onto the functional consistency of Fuse's clusters, we evaluate the functional consistency of the obtained clusters after the alignment step, when using $\gamma = 0.7$ and when α varies in $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$.

As presented in supplementary Figure C.3, Fuse achieves the highest consistency of its clusters for $\alpha = 0.8$. Interestingly, with only the sequence taken into account ($\alpha = 1.0$) our alignment method already outperforms the other aligners.

To assess whether an optimal value of α can be learnt, we use the following 2-fold cross-validation experiment. We randomly split each of the five PPI networks into

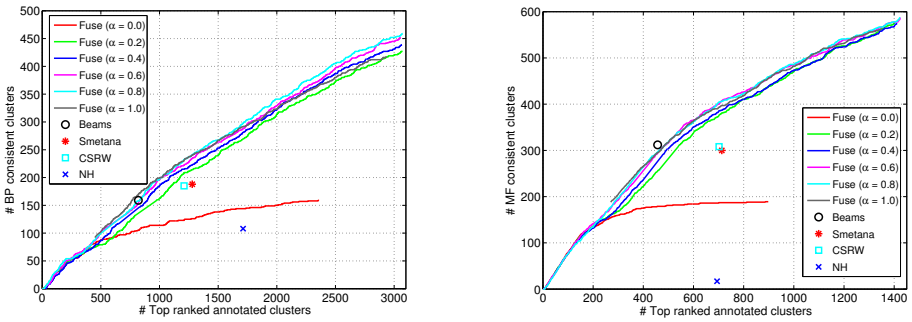


Figure C.3: Functional consistency of Fuse's clusters for different values of α. For Fuse's protein clusters with different values of α, we plot the cumulative number of annotated clusters containing proteins from all 5 species (x-axis) against the number of them that are functionally consistent (y-axis). Clusters are ranked according to the scores computed as the sum of association scores between their proteins. We also report the total number of annotated and consistent clusters for the competing aligners (points in the plots). Biological process (BP) and molecular function (MF) annotations are considered separately in the left-hand-side and right-hand-side panels, respectively.

two sub-networks having same numbers of nodes. For each PPI network, one sub-network is assigned to the training set, and the other one is assigned to the testing set. Then, we learn on the training set the value of α (from 0 to 1 in increments of 0.1) that results in the largest number of functionally consistent clusters, and use this predicted value to align the corresponding testing set. To measure how good this alignment is in comparison to alignments that used other values of α, we also align the testing set using the other values of α, and report if the learnt value of α was optimal (i.e., if it lead to the largest number of functionally consistent clusters). We also compute the percentage of functionally consistent clusters that are found in the testing set when using the learnt value of α with respect to the number of functionally consistent clusters that are found in the optimal alignment. We repeat this procedure 100 times to account for the randomness in the experiments. Hence, as we test 11 different values of α, we generated and evaluated 2,200 alignments.

We obtained the following results. First, as presented in the top panels of supplementary Figure C.4, the learnt values of α are optimal in only 10% of the experiments when functional consistency is based on GO biological process (BP) annotations and in only 16% of the experiments when functional consistency is based on GO molecular function (MF) annotations. In comparison, setting $\alpha = 0.9$ is optimal in 18% of the experiments according to BP and in 24% of the experiments according to MF. Second, as presented in the bottom panels of supplementary Figure C.4, using the learnt values of α and using a fixed value of α between 0.1 and 0.9 (as α=0 and α=1 never result in the largest number of functionally consistent clusters)

both lead to comparable functional consistencies. We find, on average, 94.2 - 95.4% of clusters to be BP consistent and 94.2 – 96.9% of clusters to be MF consistent, over clusters obtained for all values of α (percentage with respect to the optimal, i.e., most functionally consistent alignment). The lowest functional consistency is observed when α is set to 0, which finds on average 44.2% of the optimal number of BP consistent clusters and 37.4% of the optimal number of MF consistent clusters.

Overall, learning the values of α on training sets does not result in functionally more consistent alignments than when using a fixed value of α in between 0.1 and 0.9.

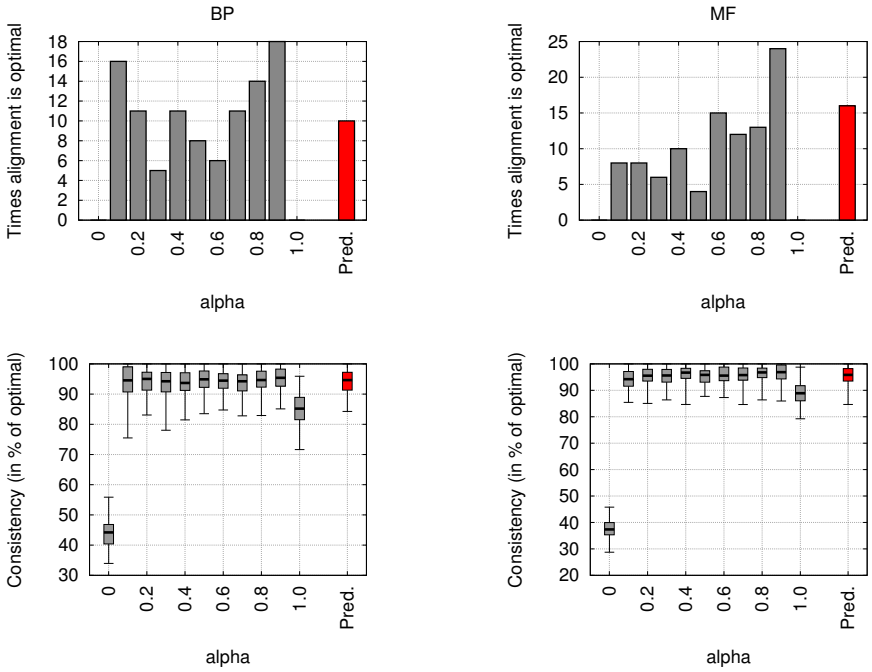


Figure C.4: 2–fold cross validation of α. *Top:* The red bars in the charts present the percentage of the experiments for which using the values of α that are learnt on the training sets to align the corresponding testing sets result in the largest numbers of functionally consistent clusters. The grey bars in the charts present the same, but when using a fixed values of α. *Bottom:* The red box-and-whiskers present the minimum, the first-quartile, the median, the third quartile and the maximum number of functionally consistent clusters that are uncovered on the testing sets when using the values of α that are learnt on the corresponding training sets. The numbers of functional consistency are presented as percentages of the maximum number of consistent clusters that are found on the testing sets when α varies from 0 to 1. In the left panels, functional consistency is based on GO biological process (BP) annotations while in the right panels it is based on GO molecular function (MF) annotations.

C.2 Study of different alignment orderings

In the alignment step of Fuse, different ordering strategies can be used to merge the networks. We consider two such strategies: 1) "min-size", which merges networks from the smaller towards the larger one; and 2) "Hierarchical", which merges networks according to the phylogenetic tree constructed from the pairwise alignments of the networks. To evaluate the influence of the ordering strategy, we compare the functional consistency of the obtained clusters after the alignment step, when Fuse is set to use either min-size, or hierarchical ordering (γ and α are set to 0.7 and 0.8, respectively).

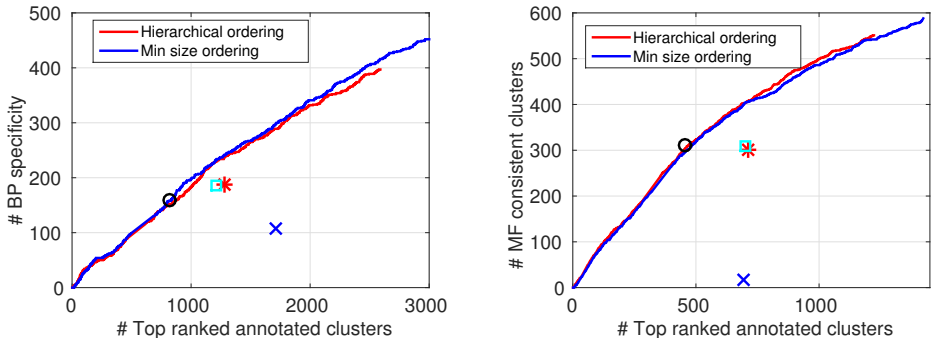


Figure C.5: Functional consistency of Fuse's clusters when using different alignment step orderings. For Fuse's protein clusters with different ordering strategies, we plot the cumulative number of annotated clusters containing proteins from all 5 species (x-axis) against the number of them that are functionally consistent (y-axis). The clusters are ranked according to the scores computed as the sum of association scores between their proteins. We also report the total number of annotated and consistent clusters for the competing aligners (points in the plots). Biological process (BP) and molecular function (MF) annotations are considered separately in the left-hand-side and right-hand-side panels, respectively.

As presented in supplementary Figure C.5, Fuse achieves the highest functional consistency when merging from the smallest network towards the largest one.

C.3 Permutation test for determining p-values of predicted associations

To determine the statistical significance of NMTF predicted associations, we use the following permutation test. We perform 200 NMTF runs, each time with a different random permutation of relation matrices. In each run, for each pair of species, i and j, we permute the entries in the corresponding relation matrix $\mathbf{R}_{ij}$ while keeping

the same distribution of degrees of proteins in species i and j. After factorizing the permuted relation matrices, we infer predictions from each reconstructed matrix $\hat{\mathbf{R}}_{ij}$. We define the p-value of a predicted association as a fraction of NMTF runs in which this association was observed.

As presented in supplementary Figure C.6, filtering associations according to p-value results in filtered associations having lower functional consistencies in comparison with the non-filtered ones.

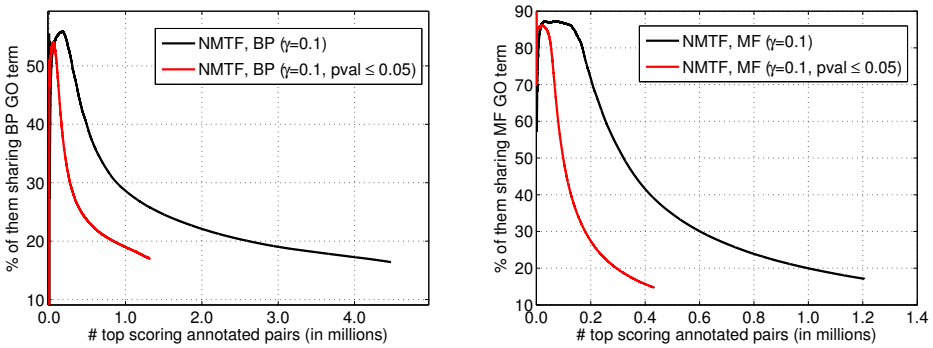


Figure C.6: Cluster BP (left panel) and MF (right panel) consistency of all GNMTF–predicted protein associations (black line) and filtered GNMTF–predicted associations with p-value ≤ 0.05 (red line).

C.4 Transferring annotations across species

To demonstrate the advantage of Fuse over the previous aligners in enabling new biological findings, we focus on clusters that cannot be identified based only on sequence similarity. Namely, NMTF can predict new associations between proteins that are not related by sequence similarities (i.e., whose sequence similarity is not significant; e-value ≥ 1). These associations can contribute to the identification of functionally consistent clusters that cannot be identified by the previous aligners, as the previous aligners cannot predict new protein associations that are not sequence-based. Furthermore, these predicted clusters could be used for transfer of annotations across proteins of different species that are not sequence-related.

To identify such clusters, we perform the following procedure. After the alignment step (for the best scoring parameters $\gamma = 0.7$, $\alpha = 0.8$), we obtain the list of clusters. For each obtained cluster, we construct an induced subgraph over its proteins in the k-partite graph composed only on sequence similarity links. We identify all the subgraphs (clusters) that are disconnected in the k-partite sequence-based graph, i.e., we identify all the clusters whose number of sequence-based links

is $< n-1$, where n is the number of proteins in the subgraph (cluster). It is evident that without NMTF predicted associations, it would not be possible to identify such clusters.

The list of 474 clusters with predicted associations is presented in the supplementary excel spreadsheet.In addition to the clusters, we also report their sequence-based and NMTF-based predicted protein associations. We further focus on clusters covering all 5 species and on their biological validation through literature. We use CiteXplore tool [290] to search for articles and biological concepts common for all 5 proteins. Due to the lack of annotations for many proteins, we validate only 5 clusters (out of 18 clusters covering all 5 species). An example of a validated cluster includes the following proteins: HPS6 (human), HPS6 (mouse), SEC72 (yeast), ABU-1 (worm) and NIMC2 (fly), where, the first three and the last two proteins are sequence-related. It is evident that if the alignment was based only on sequence scores, the identified clusters would cover either three or two species. However, due to the predicted association between proteins HPS6 (mouse) and ABU-1 (worm), Fuse was able to identify this cluster covering all 5 species. We validate this cluster through literature curation. We observe the following: protein HPS6, located in the endoplasmic reticulum, belongs to the family of Hermansky-Pudlak syndrome (HPS) proteins whose role has been shown in primary immunodeficiency disorders, i.e., a group of disorders of the immune system [291]; protein ABU-1, an integral component of the endoplasmic reticulum membrane, has been implicated in the normal physiology of immune defence [292]; protein SEC72, also located in the endoplasmic reticulum, has been shown to be involved in a mechanism protecting immunity of a virus-infected yeast [293]; protein NIMC2 belongs to a Nimrod gene superfamily that is an important component of the innate immune response [294]. Thus all 5 proteins play important roles in immune responses of the five species and the first four are located in the endoplasmic reticulum.

The non-validated clusters could be used for transfer of annotations from annotated to unannotated proteins in the clusters. For example, in the cluster covering proteins, BTLA (human), BTLA (mouse), ARN2 (yeast), ORK1 (fly) and ABU-7 (worm) (see supplementary excel spreadsheet), the first four proteins are sequence-related, and they are annotated to be located in the integral component of membrane (GO:0016021); the last protein, ABU-7 (worm), is predicted to be a part of the cluster due to its predicted association with protein BTLA (mouse), and it is not annotated for cellular component. Hence, by transferring annotation to this protein, we can postulate that ABU-7 (worm) is located in the membrane.

Appendix D

Fusing network layers on Grassmann manifold

D.1 Artificial Multiplex Networks

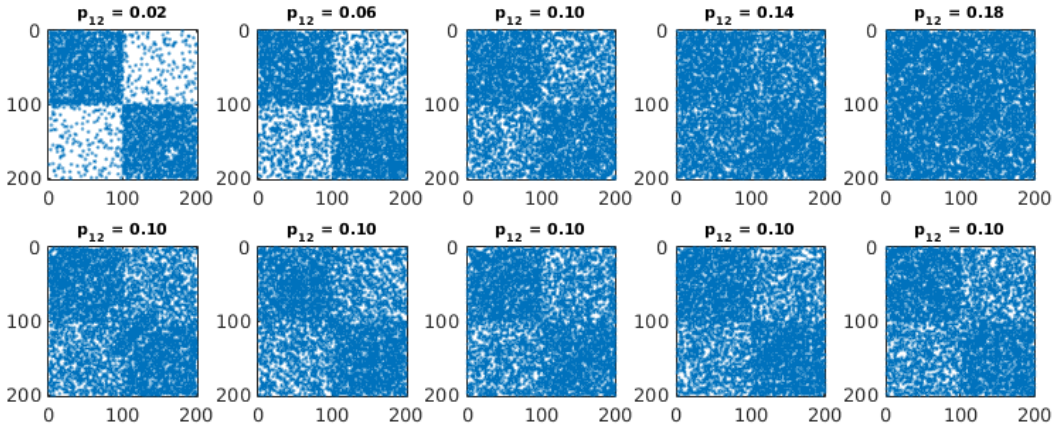


Figure D.1: *SYNTH-N*: Spy-plots of five different 2–layer artificial multiplex networks used in our experiments to demonstrate the effect of *noise* on the performance of our methods. Clusters in the top and bottom layer have fixed within-community probability, $p_{11} = 0.3, p_{22} = 0.3$ across all five networks. The between-community probability of the second layer is also fixed, $p_{12} = 0.10$ across networks, whereas, it varies in the first layer, i.e., $p_{12} = \{0.02, 0.06, 0.10, 0.14, 0.18\}$.

D.2 Derivation of update rules for NF-CCE algorithm

We use the same mathematical notation as in the main manuscript.

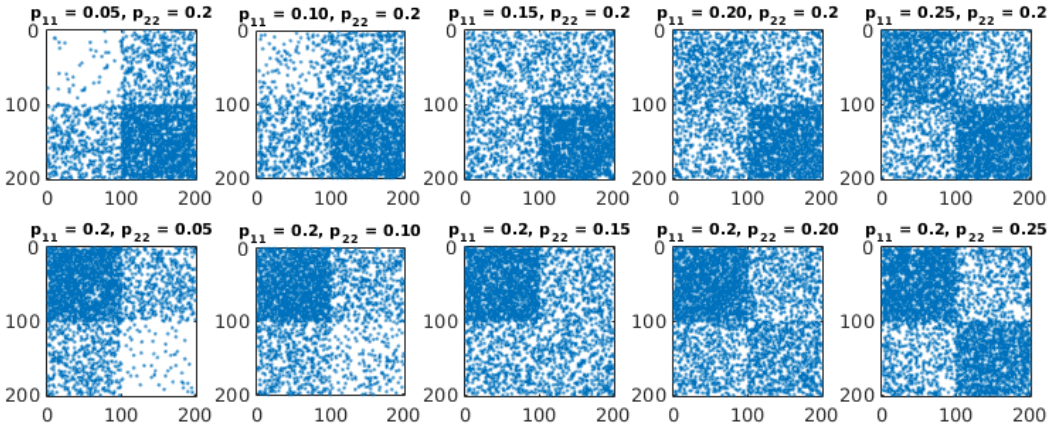


Figure D.2: *SYNTH-C*: Spy-plots of five different 2-layer artificial multiplex used in our experiments to demonstrate the ability of our methods to integrate complementary layers. Top and bottom layer of each of the five multiplex networks is generated to represent different, *complementary* information about clusters. In the first layer, the within-community probability of the first cluster varies across different networks, i.e., $p_{11} = \{0.05, 0.10, 0.15, 0.20, 0.25\}$, while the within-community probability of the second cluster is fixed across layers, $p_{22} = 0.2$. In the second layer, the information is reversed. Namely, the first cluster has a fixed within-community probability, $p_{22} = 0.2$, whereas the within-community probability of the second cluster varies across networks, i.e., $p_{11} = \{0.05, 0.10, 0.15, 0.20, 0.25\}$. The level of noise between clusters, i.e., between-community probability, is fixed in all layers and across all five multiplex networks.

Collective SNMF

Step 1: Optimization problem for each layer (SNMF):

$$\forall i \in \{1, \ldots, N\}, \quad \mathcal{J} = \min_{\mathbf{H}^{(i)} \geq 0} \parallel \mathbf{A}^{(i)} - \mathbf{H}^{(i)}\mathbf{H}^{(i)T} \parallel_F^2 \tag{D.1}$$

The update rule for minimizing $\mathcal{J}$ is:

$$\mathbf{H}_{jk}^{(i)} \leftarrow \mathbf{H}_{jk}^{(i)} \circ \frac{\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\right]_{jk}} \tag{D.2}$$

Step 2: Collective optimization problem (CSNMF):

$$\min_{\mathbf{H} \geq 0} \mathcal{J} = \min_{\mathbf{H} \geq 0} \sum_{i=1}^{N} \underbrace{\parallel \mathbf{A}^{(i)} - \mathbf{H}\mathbf{H}^T \parallel_F^2}_{\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})} + \alpha \sum_{i=1}^{N} \underbrace{(k - tr(\mathbf{H}\mathbf{H}^T\mathbf{H}^{(i)}\mathbf{H}^{(i)T}))}_{\mathcal{J}_c^{(i)}(\mathbf{H};\mathbf{H}^{(i)})} \tag{D.3}$$

The derivative of $\mathcal{J}$ with respect to $\mathbf{H}$ is:

$$\nabla_{\mathbf{H}}\mathcal{J} = -\sum_{i=1}^{N} \underbrace{4\mathbf{A}^{(i)}\mathbf{H}}_{[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^-} + \sum_{i=1}^{N} \underbrace{4\mathbf{H}\mathbf{H}^T\mathbf{H}}_{[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^+} - 2\alpha \sum_{i=1}^{N} \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H} \tag{D.4}$$

The non-negative parts of the *normal gradient*, $\widetilde{\nabla}_{\mathbf{H}}\mathcal{J} = \nabla_{\mathbf{H}}\mathcal{J} - \mathbf{H}\mathbf{H}^T\nabla_{\mathbf{H}}\mathcal{J}$ can be written as follows:

$$[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^- = \Big(\sum_{i=1}^{N} \mathbf{A}^{(i)} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\Big)\mathbf{H}$$
$$[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^+ = \mathbf{H}\mathbf{H}^T\Big(\sum_{i=1}^{N} \mathbf{A}^{(i)} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\Big)\mathbf{H}$$

Using the natural gradient, the update rule for updating elements of $\mathbf{H}$ can be obtain from:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \circ \frac{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^-_{jk}}{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^+_{jk}} \tag{D.5}$$

Introducing the average adjacency matrix as: $\mathbf{A}_{avg} = \sum_{i=1}^{N} \mathbf{A}^{(i)} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}$, the update rule for minimizing objective function in Equation D.3 can be written as:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \circ \frac{\big[\mathbf{A}_{avg}\mathbf{H}\big]_{jk}}{\big[\mathbf{H}\mathbf{H}^T\mathbf{A}_{avg}\mathbf{H}\big]_{jk}} \tag{D.6}$$

which is of the same mathematical form as the update rule for SNMF (Equation D.2).

Collective PNMF

Step 1: Optimization problem for each layer (PNMF):

$$\forall i \in \{1, \ldots, N\}, \quad \mathcal{J} = \min_{\mathbf{H}^{(i)} \geq 0} \parallel \mathbf{A}^{(i)} - \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)} \parallel_F^2 \tag{D.7}$$

The update rule for minimizing $\mathcal{J}$ is:

$$\mathbf{H}^{(i)}_{jk} \leftarrow \mathbf{H}^{(i)}_{jk} \circ \frac{\left[\mathbf{A}^{(i)}\mathbf{A}^{(i)T}\mathbf{H}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{A}^{(i)T}\mathbf{H}^{(i)}\right]_{jk}} \tag{D.8}$$

Step 2: Collective optimization problem (CPNMF):

$$\min_{\mathbf{H}\geq 0} \mathcal{J} = \min_{\mathbf{H}\geq 0} \sum_{i=1}^{N} \underbrace{\parallel \mathbf{A}^{(i)} - \mathbf{H}\mathbf{H}^T\mathbf{A}^{(i)} \parallel_F^2}_{\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})} + \alpha \sum_{i=1}^{N} \underbrace{(k - tr(\mathbf{H}\mathbf{H}^T\mathbf{H}^{(i)}\mathbf{H}^{(i)T}))}_{\mathcal{J}_c^{(i)}(\mathbf{H};\mathbf{H}^{(i)})} \tag{D.9}$$

The derivative of $\mathcal{J}$ with respect to $\mathbf{H}$ is:

$$\begin{aligned}\nabla_{\mathbf{H}}\mathcal{J} = &-\sum_{i=1}^{N} \underbrace{4\mathbf{A}^{(i)}\mathbf{A}^{(i)T}\mathbf{H}}_{[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^-} + \sum_{i=1}^{N} \underbrace{2\mathbf{A}^{(i)}\mathbf{A}^{(i)T}\mathbf{H}\mathbf{H}^T\mathbf{H} + 2\mathbf{H}\mathbf{H}^T\mathbf{A}^{(i)}\mathbf{A}^{(i)T}\mathbf{H}}_{[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^+} \\ &- 2\alpha \sum_{i=1}^{N} \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\end{aligned} \tag{D.10}$$

The non-negative parts of the *normal gradient*, $\widetilde{\nabla}_{\mathbf{H}}\mathcal{J} = \nabla_{\mathbf{H}}\mathcal{J} - \mathbf{H}\mathbf{H}^T\nabla_{\mathbf{H}}\mathcal{J}$ can be written as follows:

$$\begin{aligned}[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^- &= \Big(\sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{A}^{(i)T} + \alpha\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\Big)\mathbf{H} \\ [\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^+ &= \mathbf{H}\mathbf{H}^T\Big(\sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{A}^{(i)T} + \alpha\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\Big)\mathbf{H}\end{aligned}$$

Similar to CSNMF, we introduce an average adjacency matrix as $\mathbf{A}_{avg} = \sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{A}^{(i)T} + \alpha\mathbf{H}^{(i)}\mathbf{H}^{(i)T}$. The update rule for $\mathbf{H}$ for minimizing the objective function $\mathcal{J}$ in Equation D.9 is as follows:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \circ \frac{\left[\mathbf{A}_{avg}\mathbf{H}\right]_{jk}}{\left[\mathbf{H}\mathbf{H}^T\mathbf{A}_{avg}\mathbf{H}\right]_{jk}} \tag{D.11}$$

which again has the same mathematical form as the update rule for SNMF (Equation D.2).

Collective SNMTF

Step 1: Optimization problem for each layer (SNMTF):

$$\forall i \in \{1, \ldots, N\}, \quad \mathcal{J} = \min_{\mathbf{H}^{(i)} \geq 0, \mathbf{S}^{(i)} \geq 0} \parallel \mathbf{A}^{(i)} - \mathbf{H}^{(i)}\mathbf{S}^{(i)}\mathbf{H}^{(i)T} \parallel_F^2 \tag{D.12}$$

The update rules for minimizing $\mathcal{J}$ are:

$$\mathbf{H}_{jk}^{(i)} \leftarrow \mathbf{H}_{jk}^{(i)} \circ \frac{\left[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\right]_{jk}} \tag{D.13}$$

$$\mathbf{S}_{jk}^{(i)} \leftarrow \mathbf{S}_{jk}^{(i)} \circ \frac{\left[\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\right]_{jk}}{\left[\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\right]_{jk}} \tag{D.14}$$

Step 2: **Step 2**: Collective optimization problem (CSNMTF):

$$\min_{\mathbf{H} \geq 0} \mathcal{J} = \min_{\mathbf{H}} \sum_{i=1}^{N} \underbrace{\parallel \mathbf{A}^{(i)} - \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^{T} \parallel_F^2}_{\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)},\mathbf{S}^{(i)})} + \alpha \sum_{i=1}^{N} \underbrace{(k - tr(\mathbf{H}\mathbf{H}^{T}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}))}_{\mathcal{J}_c^{(i)}(\mathbf{H};\mathbf{H}^{(i)})} \tag{D.15}$$

The derivative of $\mathcal{J}$ with respect to $\mathbf{H}$ is:

$$\nabla_{\mathbf{H}}\mathcal{J} = -\sum_{i=1}^{N} \underbrace{4\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}}_{[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^-} + \sum_{i=1}^{N} \underbrace{4\mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^{T}\mathbf{H}\mathbf{S}^{(i)}}_{[\nabla_{\mathbf{H}}\mathcal{J}^{(i)}(\mathbf{H};\mathbf{A}^{(i)})]^+} - 2\alpha \sum_{i=1}^{N} \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H} \tag{D.16}$$

The non-negative parts of the *normal gradient*, $\widetilde{\nabla}_{\mathbf{H}}\mathcal{J} = \nabla_{\mathbf{H}}\mathcal{J} - \mathbf{H}\mathbf{H}^{T}\nabla_{\mathbf{H}}\mathcal{J}$ can be written as follows:

$$[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^- = \sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}$$

$$[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^+ = \mathbf{H}\mathbf{H}^{T}\Big(\sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\Big)$$

The update rule for $\mathbf{H}$ for minimizing the objective function in Equation D.15 is as follows:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \circ \frac{\big[\sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)} + \frac{\alpha}{2}\sum_{i=1}^{N} \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\big]_{jk}}{\big[\mathbf{H}\mathbf{H}^{T}\big(\sum_{i=1}^{N} \mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)} + \frac{\alpha}{2}\sum_{i=1}^{N} \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\big)\big]_{jk}} \tag{D.17}$$

Collective Ssemi-NMTF

Step 1: Optimization problem for each layer (Ssemi-NMTF):

$$\forall i \in \{1, \ldots, N\}, \quad \mathcal{J} = \min_{\mathbf{H}^{(i)} \geq 0, \mathbf{S}^{(i)}} \parallel \mathbf{A}^{(i)} - \mathbf{H}^{(i)}\mathbf{S}^{(i)}\mathbf{H}^{(i)T} \parallel_F^2 \tag{D.18}$$

where $\mathbf{S}^{(i)}$ is allowed to have both positive and negative elements.

The update rules for minimizing $\mathcal{J}$ are:

$$\mathbf{H}_{jk}^{(i)} \leftarrow \mathbf{H}_{jk}^{(i)} \circ \frac{\Big[\big[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\big]^{+} + \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\big[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\big]^{-}\Big]_{jk}}{\Big[\big[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\big]^{-} + \mathbf{H}^{(i)}\mathbf{H}^{(i)T}\big[\mathbf{A}^{(i)}\mathbf{H}^{(i)}\mathbf{S}^{(i)}\big]^{+}\Big]_{jk}} \tag{D.19}$$

$$\mathbf{S}^{(i)} \leftarrow \big(\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\big)^{-1}\mathbf{H}^{(i)T}\mathbf{A}^{(i)}\mathbf{H}^{(i)}\big(\mathbf{H}^{(i)T}\mathbf{H}^{(i)}\big)^{-1} \tag{D.20}$$

Step 2:Collective optimization problem (CSsemi-NMTF):

The objective function is the same as for CSNMTF (Equation D.15), however, in the derivation of update rules, the terms having $\mathbf{S}^{(i)}$ matrix needs to be separated into the non-negative parts.

The non-negative parts of the *normal gradient*, $\widetilde{\nabla}_{\mathbf{H}}\mathcal{J} = \nabla_{\mathbf{H}}\mathcal{J} - \mathbf{H}\mathbf{H}^{T}\nabla_{\mathbf{H}}\mathcal{J}$ can be written as follows:

$$\begin{aligned}
[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^{-} &= \sum_{i=1}^{N}[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}]^{+} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H} \\
&\quad + \mathbf{H}\mathbf{H}^{T}[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}]^{-} \\
[\widetilde{\nabla}_{\mathbf{H}}\mathcal{J}]^{+} &= \sum_{i=1}^{N}[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}]^{-} + \frac{\alpha}{2}\mathbf{H}\mathbf{H}^{T}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H} \\
&\quad + \mathbf{H}\mathbf{H}^{T}[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}]^{+}
\end{aligned}$$

The update rule for $\mathbf{H}$ for minimizing the objective function in Equation D.15 is as follows:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\Big[\sum_{i=1}^{N} \big[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\big]^{+} + \mathbf{H}\mathbf{H}^{T}\big[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\big]^{-} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\Big]_{jk}}{\Big[\sum_{i=1}^{N} \big[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\big]^{-} + \mathbf{H}\mathbf{H}^{T}\Big(\big[\mathbf{A}^{(i)}\mathbf{H}\mathbf{S}^{(i)}\big]^{+} + \frac{\alpha}{2}\mathbf{H}^{(i)}\mathbf{H}^{(i)T}\mathbf{H}\Big)\Big]_{jk}} \tag{D.21}$$

D.3 Parameter estimation

The state-of-the-art methods, in our comparison study, require a parameter fitting. To make the fair comparison between state-of-the-art methods and our methods, we apply the following parameter estimation strategy. For each of the methods requiring user-defined parameter, we test the method for the values of parameter given in the predefined range, and choose the optimal value of the parameter that leads to the best clustering performance in terms of *Purity*, NMI, *ARI* and R_{avg} (in the case of biological networks). The values of parameters of different methods applied on different real-world networks are summarized in Table D.1.

Table D.1: Optimal parameters for (from left to right): PMM, SC-ML, LMF, Graph-Fuse, CSNMF, CPNMF, CSNMTF, CSsemi-NMTF applied on real-world multi-layer networks (from top to bottom): *CiteSeer*, *CoRA*, *MPD*, *SND*, *WBN*, *WTN*, *HBN* and *YBN*

.

	PMM (l)	SC-ML (α)	LMF (α)	GF (λ)	CSNMF (α)	CPNMF (α)	CSNMTF (α)	CSsNMTF (α)
CiteSeer	10	0.9	1,000,000	0.1	0.05	0.45	0.4	0.2
CoRA	6	0.3	0.000001	0.00001	0.15	6.5	0.7	1.0
MPD	7	0.5	10.0	1.0	0.3	0.7	0.05	0.3
SND(o)	4	0.10	10.0	0.001	0.05	0.25	0.05	0.35
SND(s)	3	0.4	1,000	1.0	0.6	0.5	0.7	0.2
WBN	14	0.1	1,000	0.000001	0.15	0.05	0.8	0.65
WTN	7	0.2	0.00001	0.00001	0.35	0.55	1.0	0.65
HBN	102	0.3	0.01	0.01	0.01	0.00001	0.01	0.001
YBN	100	0.4	0.01	0.1	0.01	0.000001	0.001	0.001
FBN	102	0.5	0.1	0.001	0.01	0.01	0.1	1.0